Departamento de Informática

PROYECTO FIN DE CARRERA

AUDITORIA Y CONTROL EN ENTORNOS BAJO ORACLE 11G

Autor: David García Bastanchuri

Tutor: Miguel Ángel Ramos González

Leganés, Junio de 2013

Título: AUDITORIA Y CONTROL EN ENTORNOS BAJO ORACLE 11G
Autor: David García Bastanchuri
Director: Miguel Ángel Ramos González

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

A mis padres y a mi hermana Elena por su constante apoyo, motivación y sobre todo por la paciencia que han tenido.

A mi hermano Ángel que constituye para mí toda una referencia tanto a nivel personal como profesional.

A Miguel Ángel Ramos por su profesionalidad, dedicación constante y aliento.

A Maite por ayudarme, comprenderme y motivarme constantemente.

Resumen

En la actualidad, la información se ha convertido en uno de los principales activos de las empresas, constituyendo las tecnologías y los sistemas relacionados con la información una de sus principales ventajas estratégicas. El sistema gestor de bases de datos relacionales Oracle 11g, representa, en muchas ocasiones, uno de los pilares sobre los que se desarrollan y construyen estos sistemas de información, por lo que, el establecimiento de un control exhaustivo y el desarrollo de auditorías sobre esta plataforma permitirá garantizar tanto la calidad como el cumplimiento de las características de la información contenida en este tipo de sistemas: integridad, confidencialidad y disponibilidad.

Este proyecto pretende realizar una exposición de los conceptos de auditoría informática y su aplicación en entornos de bases de datos, particularizando su utilización sobre el sistema gestor de base de datos Oracle en su versión 11g. Como resultado de la aplicación práctica de los conceptos e ideas empleados, se llevará a cabo la realización de una aplicación informática, destinada a apoyar la labor del auditor en la realización de auditorías informáticas sobre este tipo de entornos, permitiéndole detectar debilidades y puntos de posible mejora.

Palabras clave: auditoría, base de datos relacional, control, Oracle 11g, sistema gestor de bases de datos relacionales.

Abstract

Today, information has become one of the main companies' assets, with technologies and systems related to information as one of its major strategic advantages. The relational database management system Oracle 11g represents, in many cases, one of the bases on which these systems develop and build information, therefore, the establishment of comprehensive controls and the development of audits over this platform, will both ensure quality and compliance with information features contained in this type of system: integrity, confidentiality and availability.

This project aims to give a presentation about computer audit concepts and their application in database environments, by specifying their use in management system Oracle database version 11g. As a result of practical application of the concepts and ideas used, the implementation of a computer application will be carried out. This application is designed to support auditor labor with audits performed on computing environments, allowing him to detect possible weaknesses and improvement points.

Keywords: audit, relational database, control, Oracle 11g, relational database management systems.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

1.1 Introducción

La información utilizada dentro del ámbito de una organización constituye un elemento clave, tanto en su gestión interna, como en el proceso de toma de decisiones estratégicas que permitan un mejor posicionamiento de dicha organización dentro del mercado, haciéndola competitiva y sostenible. Partiendo de esta base, la información debe considerarse uno de los principales activos de las empresas y por tanto debe ser gestionada de forma adecuada para conseguir los objetivos establecidos inicialmente.

En la gran mayoría de organizaciones, la información se gestiona utilizando un sistema de información que puede definirse como el conjunto de procedimientos manuales, automatizados, y de funciones dirigidas a la recogida, elaboración, almacenamiento, recuperación y distribución de la información dentro de una organización, con el objetivo de habilitar un flujo desde el punto en el que se genera hasta el destinatario de la misma.

Con la finalidad de dotar de utilidad a la información, ésta debe cumplir una serie de características: relevante, completa, precisa, actualizada y asumible económicamente por la organización. Como elementos que facilitan y constituyen un soporte para el cumplimiento de estas características, se utilizan las tecnologías de la información (*TI*) que pueden definirse como aquellas tecnologías que posibilitan la construcción y el mantenimiento de los sistemas de información.

El sistema gestor de base de datos de Oracle es utilizado como pilar para la construcción de gran cantidad de sistemas de información y su importancia y criticidad se acentúan cuando la información controlada por el mismo se erige como un recurso clave para la organización.

La importancia y criticidad del sistema gestor de base de datos Oracle hacen necesaria la obligatoriedad de establecer tanto controles exhaustivos como la realización de auditorías que permitan alcanzar los siguientes objetivos:

- Objetivos de protección de los activos y recursos: fundamentalmente la protección de la información almacenada en las bases de datos.

- Objetivos de integridad de los datos: referida al conjunto de características que deben cumplir los datos: completitud, robustez y veracidad.

- Objetivos de efectividad del sistema: relativos al cumplimento de los objetivos establecidos con antelación y a las expectativas depositadas sobre el sistema gestor de base de datos.

- Objetivos de eficiencia del sistema: asociados a la utilización del mínimo número de recursos para llevar a cabo el cumplimiento de sus funciones.

Este proyecto pretende llevar a cabo, como primer paso, una exposición de los conceptos relacionados con la auditoría informática, el control interno y su aplicación en entornos de bases de datos.

Tras realizar esta exposición de definiciones y establecer de forma rigurosa los conceptos de base de datos y sistema gestor de base de datos, se procederá a realizar un análisis de la arquitectura del sistema gestor de base de datos Oracle en su versión 11g.

El siguiente paso, será la presentación de la metodología ágil Scrum que se utilizará como referencia para el desarrollo de una aplicación informática destinada a apoyar la labor del auditor en la realización de auditorías sobre este tipo de entornos.

Una vez planteados los fundamentos de la metodología, se procederá a su aplicación mostrando el proceso de desarrollo seguido en la aplicación de apoyo a la labor del auditor.

Finalmente, se procederá a mostrar el presupuesto elaborado, las conclusiones extraídas tras la realización del proyecto y se realizará una descripción de las posibles futuras líneas de desarrollo a seguir.

1.2 Principales objetivos

Los objetivos a cubrir en este proyecto se exponen a continuación:

- Exposición de los conceptos de auditoría informática, control interno y su particularización sobre los entornos de bases de datos.

- Establecimiento de las bases de la arquitectura del sistema gestor de base de datos Oracle en su versión 11g con la finalidad de desarrollar una aplicación adaptada a las necesidades reales que se plantean en la realización de auditorías sobre este tipo de entornos.

- Aplicación de la metodología Scrum en el desarrollo de la aplicación de apoyo a la labor del auditor denominada *Automatic Audit System Oracle 11g* (*AAS11*[1]).

- Exposición del proceso de desarrollo, elaboración de un presupuesto, establecimiento de conclusiones y propuesta de futuras líneas de desarrollo y ampliación.

- La aplicación desarrollada pretende alcanzar los siguientes subobjetivos:

    - Asistencia en la labor de auditoría sobre sistemas de información basados en el sistema gestor de bases de datos Oracle 11g.
    - Presentar una interfaz amigable que permita la utilización simultánea de la aplicación desarrollada a usuarios con diferentes grados de conocimiento sobre la materia.
    - Obtención de un informe de monitorización relativo al estado del sistema gestor de base de datos.
    - Permitir el despliegue de la aplicación en entornos distribuidos utilizando el modelo cliente-servidor.

[1] Acrónimo escogido para denominar a la aplicación desarrollada.

1.3 Fases del desarrollo

Este proyecto se descompone en las siguientes fases:

- Exposición de conceptos de auditoría y control interno.
- Detalle del proceso de auditoría en entornos de bases de datos.
- Arquitectura del sistema gestor de bases de datos Oracle 11g.
- Fundamentos de la metodología de desarrollo ágil Scrum.
- Exposición del proceso de desarrollo de la aplicación *Automatic Audit System Oracle 11g* (*AAS11*).
- Elaboración de un presupuesto para el desarrollo de la aplicación.

La correlación y dependencia de estas fases se expone en la siguiente figura:

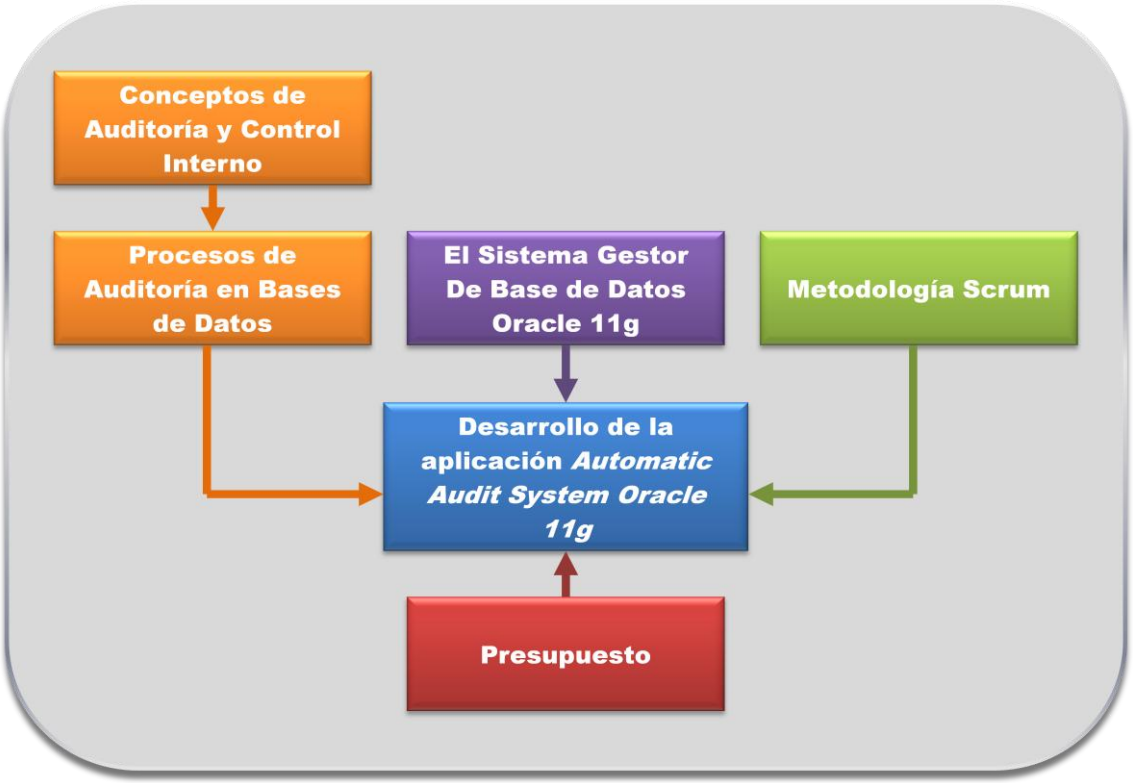


Figura 1. Correlación y dependencia de las fases del proyecto.

1.4 Medios empleados

Los medios empleados para la realización de este proyecto son los siguientes:

- Editor de textos: Microsoft Word 2007.
- Editor de imágenes: Microsoft Power Point 2007.
- Herramientas empleadas en la elaboración de la aplicación *Automatic Audit System Oracle 11g*:
    - Herramientas de desarrollo:
        - Java Development Kit: JDK 1.7.0.
        - Entorno integrado de desarrollo: Spring tool suite 3.1.0 RELEASE.
        - Entorno integrado de desarrollo de apoyo: Eclipse Juno Service Release 3.
        - Framework de desarrollo: Spring versión 3.1.2.
    - Herramientas para desarrollo del interfaz web:
        - XHTML: versión 1.0.
        - Java Server Pages (JSP): con JSTL 1.2.
        - Framework de plantillas web: Tiles 2.2.2.
        - Navegador web: Google Chrome versión 25.0.1364.172.
    - Herramienta de gestión y construcción de proyectos:
        - Apache Maven 3.0.4.
    - Servidor de aplicaciones:
        - Glassfish Server Open Source Edition 3.0.1 (build 22).
- Herramientas de generación de diagramas:
    - Diagramas de casos de uso, diagramas de clases y diagramas de secuencia: Microsoft Office Visio 2007.
    - Modelo conceptual y modelo lógico de datos: Oracle SQL Developer Data Modeler (3.1.1.703).

1.5 Estructura de la memoria

La memoria de este proyecto ha sido dividida en 8 capítulos cuyo resumen se presenta a continuación:

- En el capítulo 1 "*Introducción y Objetivos*" se realizará una introducción al proyecto, presentando las motivaciones, estructura y los objetivos que se pretenden alcanzar.

- En el capítulo 2 "*Auditoría Informática*" se realizará una definición de control interno y auditoría informática y se establecerán tanto las funciones como la relación que existe entre estas definiciones.

- En el capítulo 3 "*Auditoría de Bases de Datos*" se realizará una introducción a los conceptos de base de datos y sistema gestor de base de datos. Tras esto, se procederá a llevar a cabo un resumen del marco legal vigente y finalmente, con el objetivo de mostrar la complejidad de este tipo de entornos, se procederá a realizar una revisión del conjunto de objetivos de control a establecer durante el ciclo de vida de una base de datos.

- En el capítulo 4 "*El Sistema Gestor de Base de Datos Oracle 11g*" se llevará a cabo una exposición de su arquitectura, con el objetivo de definir, en capítulos posteriores, un procedimiento de auditoría que permita realizar un análisis lo más exhaustivo posible de todo el sistema, con la finalidad de detectar posibles debilidades y puntos de mejora.

- En el capítulo 5 "*Metodología Scrum*" se procede a realizar una descripción de la metodología ágil que se empleará en el desarrollo de la aplicación, destacando el origen de la metodología, las fases en las que se fundamenta, el equipo de trabajo que interviene habitualmente, los diagramas y las herramientas que se utilizan.

- El capítulo 6 "*Desarrollo de la Aplicación Automatic Audit System Oracle 11g*" constituye la síntesis del esfuerzo de elaboración de la aplicación *AAS11*. En este capítulo se recogerá el proceso de desarrollo del sistema implementado.

- En el capítulo 7 "*Presupuesto*" se describirá, de forma detallada, el presupuesto elaborado para afrontar el proyecto.

- En el capítulo 8 "*Conclusiones*" se expondrán las conclusiones del proyecto, las dificultades encontradas en su realización y se presentarán posibles futuras líneas de ampliación y mejora relativas a la aplicación AAS11.

Capítulo 2

Auditoría Informática

2.1 Introducción

La información, que es tratada dentro del ámbito de una organización, supone un recurso crítico que debe ser protegido, puesto que constituye uno de los elementos clave en el proceso de toma de decisiones.

Para garantizar que la información que se maneja es exacta, completa, está disponible cuando se necesita y es sólo accesible por aquellas personas autorizadas a disponer de acceso a la misma, deben implementarse controles informáticos internos que, además de ayudar a cumplir con las exigencias legales en materias de derecho informático, aseguran que los sistemas automáticos de procesamiento de la información funcionan de acuerdo a las expectativas depositadas en los mismos.

Los escándalos contables acaecidos en la primera década del siglo XXI han provocado un aumento de la sensibilización, tanto de los gobiernos como de las organizaciones (públicas y privadas) por el control interno. La existencia de una nueva normativa al respecto (Sarbanes Oxley[2], informe COSO[3]…), las necesidades de transparencia en la gestión como un valor añadido en la empresa o la búsqueda de la eficiencia en los procesos internos han supuesto un incentivo crítico en la mejora de los mecanismos de control interno. Las consideraciones realizadas por las organizaciones a este respecto, hacen que se alcance un nuevo estadio de evolución en el que la mejora de la eficiencia y el control de sus actividades constituyen los objetivos claves a alcanzar.

[2] La ley Sarbanes-Oxely, cuyo título oficial en inglés es Sarbanes Oxely Act of 2002 Pub. L No. 107-204,116 Stat. 745 (30 de Julio de 2002), nace en Estados Unidos con el fin de monitorizar las empresas que cotizan en bolsa de valores, evitando que las acciones de las mismas sean alteradas de manera dudosa, mientras que su valor es menor. Su finalidad es evitar fraudes y riesgo de bancarrota, protegiendo al inversor.

[3] El informe COSO (Committee of Sponsoring Organizations of the treadway Commission) es un documento que contiene las principales directivas para la implantación, gestión y control de un sistema de control interno.

Si nos centramos en el control interno, uno de los aspectos claves a considerar es el control sobre la gestión de los sistemas de información, motivado por diversas razones:

- La creciente dependencia de las organizaciones y sus procesos (tanto internos como externos) respecto a sus sistemas de información.

- Derivado de lo anterior, el aumento de la complejidad de los mismos, desplegados en entornos heterogéneos y abiertos, a la vez que integrados.

- El éxito de las estrategias de externalización de la gestión de los sistemas de información, con las que la dependencia de los sistemas de información se acentúa con la dependencia de uno o varios proveedores de servicio.

- La globalización.

- La gestión de la calidad total.

Estrechamente asociada al concepto de control interno, se incorpora a las organizaciones la auditoría informática que, en origen, se establece como un apoyo a la auditoría financiera y posteriormente asume nuevas funciones derivadas, en esencia, de la ocurrencia de una serie de eventos:

- Aparición de normativa específica aplicable sobre los sistemas de la información empleados dentro del ámbito de las organizaciones y sus procesos de gestión. Los ejemplos más conocidos son la Ley Orgánica de Protección de Datos (ley Orgánica 15/1999 de 13 de Diciembre. LOPD) y la ley de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSI[4]).

- Los sistemas de comercio electrónico, tanto entre organizaciones (B2B Business to Business), como orientados a clientes finales (B2C Business to Consumer), que han impulsado la mejora de los procesos de comercialización de productos pero a la vez han abierto la puerta a nuevos riesgos derivados de la necesidad de "abrir" los sistemas de información de las organizaciones a terceros.

- El aumento de la complejidad de los sistemas de información y la dependencia de las organizaciones respecto de los mismos.

En este capítulo se realizará una definición de control interno y auditoría informática y se establecerán tanto las funciones como la relación que existe entre estas definiciones. Adicionalmente, se procederá a realizar una exposición de algunos de los controles internos establecidos en diferentes áreas de una organización, con el objetivo de suministrar una perspectiva general sobre el dominio tratado.

[4] Elaborada por la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información del Ministerio de Industria, Turismo y Comercio, en cumplimiento de lo dispuesto en el artículo 33 de la citada Ley, se aplica a todas las actividades que se realicen por medios electrónicos y tengan carácter comercial o persigan un fin económico.

2.2 Control interno informático

2.2.1 Definición

Según [PPP+08] el control interno puede definirse como "cualquier actividad o acción realizada manual y/o automáticamente para prevenir, corregir errores o irregularidades que puedan afectar al funcionamiento de un sistema para conseguir sus objetivos".

Los controles durante su diseño, desarrollo e implantación han de ser al menos completos, simples, fiables, revisables, adecuados y rentables.

El control interno informático se utiliza con el objetivo de supervisar diariamente que todas las actividades relativas a un sistema de información sean realizadas cumpliendo los procedimientos, estándares y normas fijados por la dirección de la organización y/o la dirección de informática, adicionalmente al cumplimento de los requerimientos legales.

El control interno informático permitirá asegurar que las medidas que se obtienen de los mecanismos implantados por cada responsable son correctas, válidas y por tanto aplicables.

Las labores asociadas al control interno informático suelen ser desempeñadas por un órgano dependiente de la dirección del departamento de informática y está dotado de las personas y medios materiales encargados de la consecución de los siguientes objetivos:

- Controlar que todas las actividades se realizan cumpliendo los procedimientos y normas fijados, evaluar su bondad y asegurarse del cumplimiento de las normas legales.

- Asesorar sobre el conocimiento de las normas.

- Colaborar y apoyar el trabajo de la auditoría informática, así como de las auditorías externas a la organización.

- Definir, implantar y ejecutar mecanismos y controles para comprobar el logro de los grados adecuados del servicio informático, lo cual no debe considerarse como que la implantación de los mecanismos de medida y la responsabilidad del logro de esos niveles se ubique dentro de la función de control interno, sino que cada responsable de objetivos ejercerá la labor de control sobre estos niveles, y se encargará de la implantación de los medios de medida adecuados.

- Realizar en los diferentes sistemas (centrales, departamentales, redes locales, PC, etc.) y entornos informáticos (producción, desarrollo o pruebas) el control de las distintas actividades operativas sobre:

    - El cumplimiento de procedimientos, normas y controles dictados. Merece la pena resaltar en este punto la vigilancia sobre el control de cambios y versiones de software.
    - Controles sobre la producción diaria.
    - Controles sobre la calidad y eficiencia del desarrollo y mantenimiento del software y del servicio informático.
    - Controles en las redes de comunicaciones.
    - Controles en el software de base.
    - Controles en los sistemas microinformáticos.

- La seguridad informática (su responsabilidad puede estar asignada al control interno o bien puede asignársele la responsabilidad de control dual de la misma cuando está encargada a otro órgano):

    - Usuarios, responsables y perfiles de uso de archivos, bases de datos y aplicaciones.
    - Normas de seguridad.
    - Control de información clasificada.
    - Control dual de la seguridad informática.
    - Licencias y relaciones contractuales con terceros.
    - Asesorar y transmitir cultura sobre el riesgo informático.

2.2.2 Tipos de controles

Históricamente, los objetivos de los controles informáticos se han clasificado en las siguientes categorías:

- Controles preventivos: utilizados con la finalidad de evitar la ocurrencia de un determinado suceso. Como ejemplo más habitual de la utilización de este tipo de controles se puede mencionar el establecimiento de un sistema de seguridad que impida accesos no autorizados a un sistema.

- Controles de detección: si los controles preventivos no han funcionado, intervienen este tipo de controles con el objetivo de detectar lo antes posible la ocurrencia de un determinado evento. Como ejemplo de aplicación de este tipo de controles se podrían agrupar todos aquellos elementos destinados a permitir la realización de un análisis sobre el funcionamiento del sistema como: el registro de intentos de accesos no autorizados, el registro de la actividad diaria para detectar errores u omisiones, establecimiento y análisis de ficheros de *log* para la revisión del funcionamiento, etc.

- Controles correctivos: estos controles están destinados a facilitar la vuelta a la normalidad cuando se produce una determinada incidencia. Un ejemplo claro del establecimiento y utilización de este tipo de mecanismos sería la recuperación de un fichero dañado a partir de una copia de seguridad realizada con antelación.

2.3 Auditoría informática

2.3.1 Definición

Con el objetivo de obtener una definición los más completa posible de auditoría informática, se ha llevado a cabo una recopilación de varias definiciones, realizando posteriormente un ejercicio de síntesis en el que se intentan extraer las características fundamentales de las mismas:

- Según [ANSI73] la auditoría informática consiste en una actividad destinada a determinar por medio de la investigación, la adecuación de y la adhesión a, los procedimientos establecidos, instrucciones, especificaciones, códigos y estándares, u otros requisitos aplicables contractuales o de licencia, así como la eficacia de su implantación.

- Según [Del98] se define como el conjunto de técnicas y actividades destinadas a analizar, evaluar, verificar y recomendar sobre el control, la planificación, la adecuación, eficacia y seguridad de la función informática de la organización. Adicionalmente esta definición se complementa añadiendo que la auditoría informática consiste en el análisis discontinuo de un sistema de información, o del servicio informático, a petición de la dirección para mejorar la calidad, la seguridad y la eficacia.

- Según [Alo89] es un examen metodológico del servicio informático, o de un sistema informático en particular, realizado de una forma puntual y de modo discontinuo, a instancias de la dirección, con la intención de contribuir a mejorar conceptos como la seguridad, la eficacia, y la rentabilidad del servicio o del sistema que se somete al proceso de auditoría.

- Según [Ach94] la auditoría informática es el mecanismo/proceso metodológico para valorar y evaluar la confianza que se puede depositar en los sistemas basados en las tecnologías de la información.

- Según [PPP+08] la auditoría informática es el proceso de recoger, agrupar y evaluar evidencias para determinar si un sistema de información salvaguarda los activos, mantiene la integridad de los datos, lleva a cabo eficazmente los fines de la organización y utiliza eficientemente los recursos.

De este modo, la auditoría informática sustenta y confirma la consecución de los objetivos tradicionales de la auditoría:

- Objetivos de protección de activos.

- Objetivos de gestión que abarcan, no solamente los de protección de activos, sino también los de eficacia y eficiencia.

En estas definiciones destacan varios elementos sobre los que podemos realizar un análisis, debido al contenido crítico que tienen con respecto a la función y objetivo de una auditoría sobre un sistema de información:

- La palabra "información" como sufijo en la denominación de "sistema de información" o "tecnología de la información" que aparece en la mayoría de definiciones anteriores, indica claramente el fundamento y objetivo principal: la fiabilidad de la información procesada, disponible y suministrada a través de procesos tecnológicos.

- La auditoría de sistemas de información sigue un proceso metodológico en sus revisiones y debe identificar la adecuación de los controles, su nivel de cumplimiento y, fundamentalmente, identificar los riesgos para el negocio. Es decir, la auditoría de sistemas de información no se limita a determinar que existe un determinado control, sino que va más allá: si el control ha sido diseñado adecuadamente para el objetivo que se pretende alcanzar, es eficiente (eficacia con respecto a coste), si se cumple con eficacia en relación al tiempo y, adicionalmente, si los riesgos de la utilización de tecnologías de la información relacionados con este control, están mitigados. La auditoría de un sistema de información no solo debe abarcar la parte automatizada del propio sistema sino que debe extenderse sobre cualquier elemento relacionado con el mismo, como podría ser el archivado de informes generados por la propia aplicación, control del conjunto de personas encargadas de su gestión …

- La obtención de evidencias tras el análisis y evaluación del conjunto de pruebas realizadas con la finalidad de poder emitir una opinión fundada y documentada.

La auditoría de sistemas de información distingue entre dos grandes grupos de controles en un entorno de tecnologías de la información:

- Los controles sobre las infraestructuras en las que se soportan las tecnologías.

- Los controles incluidos en las propias aplicaciones o software dedicado a la gestión de la actividad de las organizaciones.

Estos dos grupos, son extremadamente dependientes, puesto que cualquier deficiencia en uno de ellos afecta directamente al otro.

La revisión, como parte de las tareas de la auditoría de tecnologías de la información, tanto de los controles sobre las infraestructuras como de los controles incluidos en las propias aplicaciones, requiere una visión global y una aplicación de una metodología al auditor de sistemas de información, sobre la organización y la utilización que se realiza de las tecnologías de la información.

2.3.2 El auditor informático

Tradicionalmente, el término auditor ha sido asociado a un revisor de cuentas colegiado. La figura tradicional del auditor, desde un punto de vista genérico, ha asumido las funciones de:

- Evaluador de la situación económica de una organización.
- Evaluador de la eficacia y de la eficiencia en el uso de los recursos.
- Responsable del establecimiento de mecanismos y de la utilización de los medios disponibles que permitan la realización de estas evaluaciones.
- Desde el punto de vista de la informática, y particularizando el ámbito de actuación a los sistemas de información, se puede establecer que el auditor llevará a cabo las siguientes labores:
    - Análisis de los sistemas de información con el objetivo de diseñar e implantar controles que permitan realizar una evaluación posterior.
    - Revisión y examen de los controles implantados en los sistemas de información para verificar su cumplimiento. Estos controles serán diseñados de acuerdo a las instrucciones de la dirección y cumplimiento de la normativa interna, requerimientos legales, protección de la confidencialidad y cobertura ante errores y fraudes.
    - Revisión y análisis del nivel de eficacia alcanzado, de la utilidad, la fiabilidad y seguridad del sistema de información en su conjunto.
    - Informe a la dirección de la organización tanto sobre el diseño, como del resultado de la aplicación de los controles implantados, así como sobre la fiabilidad de la información suministrada.

Dada la complejidad actual de los sistemas de información, en general no será posible verificar manualmente los procedimientos informatizados establecidos que resumen, calculan y clasifican datos, por lo que se deberá emplear software de auditoría y otras técnicas que requieren el apoyo de los medios informáticos.

2.3.3 Objetivos de la auditoría informática

El objetivo general de la auditoría de sistemas de información es el de la comprobación de la fiabilidad y operatividad de las funciones desempeñadas por un sistema de información. Los objetivos de la auditoría de los sistemas de información incluyen la protección de activos, la integridad de los datos, alcance de efectividad y eficiencia.

De forma más detallada, podemos enumerar los objetivos de la auditoría informática a continuación:

- Objetivos de protección de los activos y recursos: los activos de un sistema de información incluyen el conjunto de recursos materiales (máquinas, mobiliario…), inmateriales (software, datos...), inmuebles y recursos personales (empleados y organización). Debe existir un sistema de control interno que proteja estos activos de todas las posibles amenazas y riesgos.

- Objetivos de integridad de los datos: la integridad de los datos es el conjunto de condiciones que deben cumplir los datos: completitud, robustez, pureza y veracidad; para que puedan reflejar con fidelidad la situación económico-financiera y general y otros hechos relacionados con la empresa. El sistema de control interno debe tener mecanismos que vigilen constantemente el mantenimiento de estas características.

- Objetivos de efectividad del sistema: un sistema de proceso de datos efectivo alcanza sus objetivos. En la evaluación de la efectividad hay que conocer de antemano las características y necesidades del usuario y los canales y procedimientos de decisión. La auditoría de la efectividad se puede realizar durante la fase de diseño de un sistema, o cuando el sistema está en funcionamiento después de un cierto tiempo.

- Objetivos de eficiencia del sistema: un sistema de proceso de datos eficiente utiliza el mínimo número de recursos para producir las salidas requeridas. Los recursos suelen ser escasos y caros en su operación, y además habitualmente están compartidos entre diferentes procesos. La eficiencia no debe medirse de una forma aislada sino que debe medirse considerando el conjunto de procesos y el conjunto de recursos disponibles.

2.3.4 Importancia de la auditoría

La auditoría cumple una función valiosa e independiente, puesto que aunque, a priori, no toma acciones, facilita recomendaciones y sus conclusiones deben tenerse en cuenta en el momento de la toma de una decisión. La auditoría se apoya en herramientas de análisis, verificación y exposición; conformando así elementos de juicio que permitirán determinar las debilidades y disfunciones del sistema.

La auditoría informática aporta información sobre la situación de los sistemas de información de la empresa a la alta dirección, que en algunos casos puede ser desconocida. También habilita la posibilidad de descubrir la existencia de algún tipo de delito o de fraude cometido, así como la posible falta de una planificación en el desempeño normal o ante posibles situaciones de desastre. Puede identificar la falta de una política clara de actuación, objetivos, normas, metodología y estándares adecuados para la organización en el ámbito de los sistemas de gestión de la información.

El hecho de realizar una auditoría informática es importante, debido a que las herramientas que se utilizan pueden definir o marcar la diferencia con respecto a la competencia, estableciéndose como una clara ventaja.

Una organización no puede permitir que el software y el hardware que utiliza presente falta de eficiencia puesto que va en contra de sus propios intereses. Adicionalmente, la seguridad supone un punto estratégico al que prestar especial atención.

La auditoría efectuada sobre los datos es muy recomendable, debido a que los sistemas pueden tener fallos en la información contenida y como consecuencia generar resultados erróneos.

La aparición de ciertos elementos indicativos puede utilizarse con la finalidad de determinar la necesidad de la realización de una auditoría informática. En estos casos, existe la posibilidad de que una empresa solicite la realización de una auditoría externa para esclarecer donde se localizan los fallos.

Existen diferentes síntomas indicativos de la existencia de posibles problemas. Estos síntomas pueden ser agrupados en clases:

- Existencia de desorganización y descoordinación: como consecuencia, los promedios conseguidos no se ajustan a los estimados o los objetivos previstos no coinciden con los resultados obtenidos.

- Existencia de insatisfacción del cliente en el caso de la prestación de un servicio: es decir, no se consigue cumplir las necesidades del cliente, se producen fallos en los sistemas que provocan el desconcierto del usuario o bien los hitos establecidos no son entregados en los plazos inicialmente propuestos.

- Aparición de debilidades de carácter económico o financiero.

2.4 Control interno y auditoría informática

Una vez expuestos los conceptos de control interno y auditoría informática se procede a establecer las relaciones entre ambos. La correlación existente entre ambos es alta y en muchas ocasiones es difícil establecer una distinción clara. Profesionales dedicados en la actualidad al control interno, han recibido en algún instante formación en seguridad informática, habiendo ejercido de forma previa labores de auditor. En la siguiente tabla se representan las similitudes y diferencias encontradas entre estas funciones:

	Control Interno Informático	Auditor Informático
SIMILITUDES	• Conocimientos especializados en tecnología de la información. • Verificación del cumplimiento de controles internos, normativa y procedimientos establecidos por la dirección de informática y la dirección general para los sistemas de información.	
DIFERENCIAS	• Análisis de los controles en el día a día. • Informa a la dirección del departamento de informática. • Sólo personal interno. • El alcance de sus funciones sobre el departamento de informática.	• Análisis en un momento temporal determinado. • Informa a la dirección general de la organización. • Personal interno y/o externo. • Tiene cobertura sobre todos los componentes de los sistemas de información de la organización.

Tabla 1. Relación entre control interno y auditoría informática.

El equipo de personas encargadas de desempeñar la labor de control interno evalúa de forma regular los controles establecidos, está constituido por personal perteneciente a la propia empresa y organizado en una jerarquía generalmente dependiente del departamento de informática. El auditor informático evalúa y comprueba en determinados momentos del tiempo los controles y procedimientos aplicados, manteniendo una comunicación directa con la dirección general de la organización.

2.5 Controles internos en una organización

Con el objetivo de entender la variedad de elementos a considerar dentro de una organización y el conjunto de controles a establecer, se llevará a cabo una exposición tanto de los aspectos previos a tener en cuenta a la hora de implantar un sistema de control, como del conjunto de controles relativos a cada área de aplicación considerada. Estos controles serían los que tanto el control interno informático, como la auditoría informática, deberían verificar para determinar su cumplimiento y validez.

2.5.1 Elementos previos a considerar

Con la finalidad de establecer un sistema de controles internos hay que definir una serie de elementos que permitirán su posterior aplicación:

- Gestión de sistemas de información: políticas, pautas y normas técnicas que sirvan de base para el diseño y la implantación de los sistemas de información y de los controles correspondientes.

- Administración de sistemas: controles sobre la actividad de los centros de proceso de datos y otras funciones de apoyo al sistema, incluyendo la administración de las redes.

- Seguridad: incluye las tres clases de controles fundamentales implantados en el software del sistema: integridad del sistema, confidencialidad (control de acceso) y disponibilidad.

- Gestión del cambio: separación de las pruebas y la producción a nivel de software y controles de procedimientos para la migración de programas software entre entornos.

La implantación de una política y cultura sobre la seguridad requiere que sea realizada por fases y esté respaldada por la dirección. Cada función juega un papel importante en las distintas etapas:

- Dirección de negocio o dirección de sistemas de información: estos órganos han de definir la política y/o directrices para los sistemas de información en base a las exigencias del negocio que podrán ser internas y externas.

- Dirección de informática: debe definir las normas de funcionamiento del entorno informático y de cada una de las funciones de informática mediante la creación y publicación de procedimientos, estándares, metodologías y normas, aplicables a todas la áreas de informática así como a los usuarios, que establecen el marco de funcionamiento.

- Control interno informático: ha de definir los diferentes controles periódicos a realizar en cada una de las funciones informáticas, de acuerdo al nivel de riesgo de cada una de ellas, y diseñarlos conforme a los objetivos de negocio y dentro del marco legal aplicable. Estos se plasmarán en los oportunos procedimientos de control interno y podrán ser preventivos o de detección. Realizará periódicamente la revisión de los controles establecidos de control interno informático informando de las desviaciones a la dirección de informática y sugiriendo cuantos cambios crea convenientes en los controles, así como transmitirá constantemente a toda la organización de informática la cultura y políticas del riesgo informático.

- Auditor interno/externo informático: ha de revisar los diferentes controles internos definidos en cada una de las funciones informáticas y el cumplimiento de normativa interna y externa, de acuerdo al nivel de riesgo, conforme a los objetivos definidos por la dirección de negocio y la dirección de informática. Informará a la alta dirección de los hechos observados y al detectarse deficiencias o ausencias de controles recomendarán acciones que minimicen los controles que pueden originarse.

La creación de un sistema de control informático es una responsabilidad de la gerencia y un punto destacable de la política establecida en el entorno informático.

En los siguientes puntos se indican algunos controles internos para los sistemas de información, agrupados por secciones funcionales.

2.5.2 Controles generales organizativos

- Políticas: deberán servir de base para la planificación, control y evaluación por la dirección de las actividades del departamento de informática.

- Planificación:

    - Plan estratégico de información, realizado por los órganos de la alta dirección de la empresa donde se definen los procesos corporativos y se considera el uso de las diversas tecnologías de información así como las amenazas y oportunidades de su uso o de su ausencia.
    - Plan informático, realizado por el departamento de informática, determina los caminos precisos para cubrir las necesidades de la empresa plasmándolas en proyectos informáticos.

  - Plan de emergencia ante desastres, que garantice la disponibilidad de los sistemas ante eventos.

- Estándares: que regulen la adquisición de recursos, el diseño, desarrollo y modificación y explotación de sistemas.

- Procedimientos: que describan la forma y las responsabilidades de ejecución para regular las relaciones entre el departamento de informática y los departamentos usuarios.

- Organizar el departamento de informática en un nivel suficientemente superior de estructura organizativa como para asegurar su independencia de los departamentos usuarios.

- Descripción de las funciones y responsabilidades dentro del departamento con una clara separación de las mismas.

- Políticas de personal: selección, plan de formación, plan de vacaciones, evaluación y promoción.

- Asegurar que la dirección revisa todos los informes de control y resuelva las excepciones que ocurran.

- Asegurar que existe una política de clasificación de la información para saber dentro de la organización qué personas están autorizadas y a qué información.

- Designar oficialmente la figura de control interno informático y de auditoría informática (estas dos figuras se nombrarán internamente y serán dependientes del tamaño del departamento de informática).

2.5.3 Controles sobre el desarrollo, adquisición y mantenimiento de sistemas de información

Sobre el desarrollo, la adquisición y el mantenimiento de los sistemas de información se establecerán una serie de controles cuyo punto de partida será tanto la selección como el seguimiento de la metodología de ciclo de vida a aplicar. El empleo de esta metodología podrá garantizar a la alta dirección que se alcanzarán los objetivos previamente establecidos sobre el sistema.
Algunos de los controles relacionados con la metodología se enuncian a continuación:

- La alta dirección debe publicar una normativa sobre el uso de metodología de ciclo de vida del desarrollo de sistemas y establecer revisiones periódicas sobre la misma.

- La metodología debe establecer los papeles y responsabilidades de las distintas áreas del departamento de informática y de los usuarios, así como la composición y responsabilidades del equipo de proyecto.

- Las especificaciones del nuevo sistema deben ser definidas por los usuarios y quedar escritas y aprobadas antes de que comience el proceso de desarrollo.

- Debe establecerse un estudio tecnológico de viabilidad en el cual se formulen formas alternativas de alcanzar los objetivos del proyecto acompañadas de un análisis coste-beneficio de cada alternativa.

- Cuando se seleccione una alternativa debe realizarse el plan director del proyecto. En dicho plan se deberá contemplar algún tipo de mecanismo de control de costes.

- Procedimientos para la definición y documentación de especificaciones de: diseño, de entrada, de salida, de ficheros, de procesos, de controles de seguridad, de establecimiento de trazas para su seguimiento y análisis por parte de la auditoría, etc.

- Plan de validación, verificación y pruebas.

- Estándares de prueba de programas, de prueba de sistemas.

- Los procedimientos de adquisición de software deberán seguir las políticas de adquisición de la organización y dichos productos deberán ser probados y revisados antes de su puesta en producción.

- Deberán prepararse manuales de operación y mantenimiento como parte de todo el proyecto de desarrollo o modificación de sistemas de información, así como manuales de usuario.

- Explotación y mantenimiento: el establecimiento de controles asegurará que los datos se tratan de forma consistente, exacta y que el contenido de sistemas sólo será modificado mediante la autorización adecuada. A este nivel, los controles a implantar serán relativos a:

    - Procedimientos de control de explotación.
    - Sistemas de contabilidad para asignar a usuarios los costes asociados a la explotación de un sistema de información.
    - Procedimientos para realizar un seguimiento y control de un sistema de información.

2.5.4 Controles relacionados con la explotación de los sistemas de información

Este tipo de controles estarán relacionados con todos aquellos elementos hardware y software que componen el sistema de información estableciendo tanto su política de adquisición como de explotación. A continuación, se enuncian algunos de estos controles:

- Planificación y gestión de recursos: definición del presupuesto operativo del departamento de informática, establecimiento del plan de adquisición de equipos y gestión de la capacidad de los equipos.

- Controles para usar, de manera efectiva, los recursos en ordenadores:

    - Calendario de carga de trabajo.
    - Programación de personal.
    - Mantenimiento preventivo del material.
    - Gestión de contingencias y cambios.
    - Procedimientos de facturación a usuarios.

- Procedimientos de selección del software del sistema, de instalación, de mantenimiento, de seguridad y control de cambios.

- Seguridad física y lógica:

    - Definición de un grupo de seguridad de la información, siendo una de sus funciones la administración y gestión del software de seguridad, revisar periódicamente los informes de violaciones y actividad de seguridad para identificar y resolver incidentes.
    - Controles físicos para asegurar que el acceso a las instalaciones del departamento de informática queda restringido a personas autorizadas.
    - Instalación de medidas de protección contra incendios.
    - Formación y concienciación en procedimientos de seguridad y evacuación.
    - Control de acceso restringido a ordenadores mediante la asignación de un identificador de usuario con palabra clave personal e intransferible.
    - Normativa que regule el acceso a los equipos informáticos.
    - Existencia de un plan de contingencias para el respaldo de recursos de ordenador críticos y para la recuperación de los servicios del departamento de informática.

2.5.5 Controles sobre las aplicaciones

Cada aplicación debe llevar controles incorporados para garantizar la entrada, actualización, validez, mantenimiento completo y exacto de los datos. Algunas cuestiones determinantes sobre el control de los datos son:

- Control de entrada de datos: procedimientos de conversión y de entrada, validación y corrección de datos.

- Controles de tratamientos de datos para asegurar que no se dan de alta, modifican o borran datos no autorizados para garantizar la integridad de los mismos mediante procesos no autorizados.

- Controles de salidas de datos: relativos a la corrección y adecuación de las salidas suministradas por el sistema, establecimiento de procedimientos de distribución de salidas, de gestión de errores en las salidas, etc.

2.5.6 Controles sobre determinadas tecnologías

En este punto, realizamos una distinción entre las principales tecnologías utilizadas habitualmente a nivel corporativo, enunciando las principales características a contemplar en cada una de ellas:

- Controles en sistemas de gestión de bases de datos:

    - El software de gestión de bases de datos para prever el acceso a, la estructuración de y el control de los datos compartidos deberá instalarse y mantenerse de modo tal que asegure la integridad del software, las bases de datos y las instrucciones de control que definen el entorno.
    - Definición de responsabilidades sobre la planificación, organización, dotación y control de los activos de datos, es decir, un administrador de datos.
    - Definición de procedimientos para la descripción sobre los cambios de datos, así como para el mantenimiento del diccionario de datos.
    - Controles sobre el acceso a datos y de concurrencia.
    - Controles para minimizar fallos, recuperar el entorno de las bases de datos hasta el punto de la caída y minimizar el tiempo necesario para la recuperación.
    - Controles para asegurar la integridad de los datos: fundamentalmente información de control para garantizar el correcto funcionamiento de las transacciones.

- Controles sobre la informática distribuida y redes:

    - Planes adecuados de implantación, modificación y pruebas de aceptación para la red.
    - Existencia de un grupo de control de red.
    - Controles para asegurar la compatibilidad de aplicaciones cuando la red es distribuida.
    - Identificación de todos los conjuntos de datos sensibles de la red y determinación de las especificaciones para su seguridad.
    - Existencia de inventario de todos los activos de la red.

- Procedimientos de respaldo del hardware y del software de la red.
- Existencia de mantenimiento preventivo de todos los activos.
- Establecimiento de controles de seguridad lógica: control de acceso a la red, establecimiento de perfiles de usuario.
- Procedimientos de cifrado de la información sensible que se transmite a través de la red.
- Monitorización para medir la eficiencia de la red.
- Identificación de los mensajes por una clave individual de usuario, por terminal y por número de secuencia del mensaje.
- Determinar si el equipo multiplexor/concentrador/procesador frontal/ remoto tiene lógica redundante y posibilidad de respaldo con realimentación automática en caso de fallo.
- Asegurarse de que existan trazas que puedan utilizarse con la finalidad de reconstruir archivos de datos y transacciones entre los diversos terminales.

- Controles sobre ordenadores personales y redes de área local:

  - Políticas de adquisición y utilización.
  - Normativas y procedimientos de desarrollo y adquisición de software de aplicación.
  - Procedimientos de control de software contratado bajo licencia.
  - Controles de acceso a redes, mediante palabra clave, en ordenadores personales.
  - Procedimientos de seguridad física y lógica.
  - Inventario actualizado de la totalidad de aplicaciones de la entidad.
  - Política referente a la organización y utilización de los discos duros de los equipos, así como para la nomenclatura de los archivos que contienen, y verificar que contemplan al menos: obligatoriedad de etiquetar el disco duro con el número de serie del equipo, creación de un subdirectorio en el que se almacenarán todos los archivos privados, así como creación de un subdirectorio público que contendrá todas las aplicaciones de uso común para los distintos usuarios.
  - Implantar herramientas de gestión de la red con el fin de valorar su rendimiento, planificación y control.
  - Adoptar procedimientos de control y gestión adecuados para la integridad, privacidad, confidencialidad y seguridad de la información contenida en redes de área local.
  - Establecimiento de mantenimientos preventivos, de detección y correctivos.
  - Establecimiento de un registro documental de las acciones de mantenimiento realizadas, incluyendo la descripción del problema y la solución dada al mismo.
  - Los ordenadores deberán estar conectados a equipos de continuidad (UPS[5], grupos autónomos…).

[5] Sistema de Alimentación Ininterrumpida, SAI (en inglés *Uninterruptible Power Supply*, *UPS*), es un dispositivo que puede proporcionar energía eléctrica a todos los dispositivos que tenga conectados.

- Control de acceso físico a los datos y aplicaciones: almacenamiento de backups, procedimientos de destrucción de datos e informes.
- Implantación de herramientas y utilidades de seguridad.
- Establecimiento de medidas que permitan la adecuada identificación de usuarios en operaciones de: altas, bajas y modificaciones, cambios de password y acceso a logs de sistema.
- Control de conexiones remotas in/out: *Módems*[6] y *Gateways*[7].

2.5.7 Controles de calidad

Este tipo de controles se establecen con la finalidad de garantizar la calidad tanto del proceso seguido como de los resultados obtenidos:

- Existencia de un plan general de calidad basado en el plan de la entidad a largo plazo y el plan a largo plazo de tecnología. Este plan general de calidad debe promover la filosofía de mejora continua y debe dar respuesta a preguntas básicas de "qué", "quién" y "cómo".

- Esquema de garantía de calidad: la dirección de informática debe establecer una norma que fije un esquema de garantía de calidad que se refiera tanto a las actividades de desarrollo de proyectos, como a las demás actividades de informática. Las normas deben establecer los tipos de actividades para garantizar la calidad (como revisiones, auditorías, inspecciones…) que deben ser realizadas para lograr los objetivos del plan general de calidad.

- Compatibilidad de la revisión de garantía de calidad con las normas y procedimientos habituales en las distintas funciones de informática.

- Metodología de desarrollo de sistemas[8]: la dirección de informática de la entidad debe definir e implementar normas para desarrollo de sistemas y adoptar una metodología de desarrollo de sistemas para administrar y gestionar dicho proceso en base al tipo de sistemas de cada entidad.

- Actualización de la metodología de desarrollo de sistemas respecto a cambios en la tecnología.

[6] Un módem (modulador /demodulador) es un dispositivo que sirve para enviar una señal llamada moduladora mediante otra señal llamada portadora utilizando habitualmente la línea telefónica.
[7] Dispositivo que permite interconectar redes con protocolos y arquitecturas diferentes a todos los niveles de comunicación.
[8] Podría seleccionarse una metodología de desarrollo de sistemas estandarizada como Métrica v3, RUP (Rational Unified Process) o incluso una metodología de desarrollo ágil como Scrum.

- Coordinación y comunicación: la dirección de informática debe establecer un procedimiento para asegurar la estrecha coordinación y comunicación con los usuarios de la entidad e informática. Este proceso debe hacerse mediante métodos estructurados, utilizando la metodología de desarrollo de sistemas para asegurar la obtención de soluciones de informática de calidad que cumplan con las necesidades de la entidad.

- Relaciones con proveedores que desarrollan sistemas: existencia de un proceso que asegure buenas relaciones laborales con proveedores que desarrollan sistemas para la entidad. Este proceso debe hacer que el usuario y el proveedor del sistema acuerden criterios de aceptación y administración de cambios, problemas durante el desarrollo, funciones del usuario, herramientas, software, normas y procedimientos.

- Normas de documentación de programas: existencia de normas de documentación de programas las cuales deben ser comunicadas e impuestas al personal pertinente. La metodología debe asegurar que la documentación creada durante el desarrollo del sistema o proyecto respete estas normas.

- Normas de pruebas de programas: la metodología de desarrollo de sistemas de la entidad debe incorporar normas que se refieran a los requisitos de las pruebas de programas, comprobación, documentación y retención de material, para probar cada uno de los módulos de software a ser puestos en producción.

- Normas respecto a la prueba de sistemas: la metodología de desarrollo de sistemas de la entidad debe incorporar normas que se refieran a los requisitos de las pruebas de sistemas, comprobación, documentación y retención del material, para probar de manera global el funcionamiento de cada sistema a ser puesto en producción.

- Pruebas piloto o en paralelo: la metodología de desarrollo de sistemas de la entidad debe definir las circunstancias bajo las cuales se efectúan pruebas piloto o en paralelo de programas o sistemas.

- Documentación de las pruebas de sistemas: la metodología de desarrollo de sistemas de la entidad debe establecer como parte de cada desarrollo, implementación o modificación, que se documenten los resultados de la prueba de sistemas.

- Evaluación del cumplimiento de garantía de calidad de las normas de desarrollo.

Capítulo 3

Auditoría de Bases de Datos

3.1 Introducción

En el capítulo 2 se ha establecido un punto de partida inicial, en el que se ha procedido a definir los conceptos de control interno y auditoría informática, presentando distintas propuestas de controles internos aplicables a diversas áreas de la organización. En este capítulo, se llevará a cabo una exposición detallada del proceso de auditoría aplicable sobre las bases de datos, con la finalidad de establecer los fundamentos a considerar en un proceso de auditoría sobre el sistema gestor de bases de datos relacionales Oracle en su versión 11g.

La gran difusión de los sistemas gestores de bases de datos junto con la consagración de los datos como uno de los recursos fundamentales de las empresas, han hecho que los temas relativos a su control interno y auditoría cobren cada día mayor interés. El control interno y la auditoría de bases de datos resultan pilares fundamentales para el correcto funcionamiento de una organización.

Uno de los aspectos más importantes que ha supuesto un impulso en el desarrollo de las técnicas de control interno y auditoría, ha sido la entrada en vigor de la Ley Orgánica 5/1992 de 29 de Octubre de regulación del Tratamiento Automatizado de Datos de carácter personal (LORTAD[9]) y del reglamento correspondiente, así como de su sucesora la Ley Orgánica 15/1999, de 13 de Diciembre de Protección de Datos de carácter personal (LOPD) que traspone la Directiva 95/46/CE, de 24 de Octubre, del Parlamento Europeo y del Consejo, relativa a la protección de las personas físicas en lo que respecta al tratamiento de los datos personales y a la libre circulación de los mismos.

[9] Aunque la LORTAD está derogada por la ley vigente actualmente, la LOPD, el Real Decreto 1332/94 que la desarrolla, ha permanecido en vigor hasta la llegada del reglamento que desarrolla la LOPD. El Real Decreto actualmente vigente que ha desarrollado la LOPD es el Real Decreto 1720/2007, de 21 de diciembre.

La documentación que se tomará como base para extraer el conjunto de recomendaciones a aplicar en una auditoría sobre un sistema gestor de base de datos será el estándar internacional ISO/IEC 27002 [ISO/IEC05] que establece un conjunto de normas para la práctica de la gestión de la seguridad de la información. Adicionalmente, se tomarán como fuentes de documentación el Information Technology Governance Institute[10] (ITGI) y la ISACA[11]. El COBIT (Control Objectives for Information and related Technology) establece un marco de trabajo para la gestión y el gobierno de las tecnologías de la información. Está compuesto por un conjunto de herramientas de apoyo que permite a los gestores y administradores salvar las distancias entre las necesidades de control, cuestiones técnicas y los riesgos de negocio.

Para comenzar, se realizará una introducción a los conceptos de base de datos y sistema gestor de base de datos. Tras esto, se procederá a llevar a cabo un resumen del marco legal vigente puesto que constituye uno de los pilares en los que se basa el cuerpo principal del capítulo, en el que se realizará una exposición detallada sobre el conjunto de recomendaciones actualmente aplicables en un proceso de auditoría sobre este tipo de sistemas. Finalmente, con el propósito de mostrar la complejidad de este tipo de entornos, se procederá a realizar una revisión del conjunto de objetivos de control a establecer durante el ciclo de vida de una base de datos.

3.2 Base de datos y sistema gestor de base de datos

3.2.1 Base de Datos

Con la finalidad de obtener una definición lo más completa posible del concepto "base de datos", se ha procedido a realizar una recopilación de varias definiciones:

- Según [Mar75] una base de datos está compuesta de una colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.

[10] El Information Technology Governance Institute (ITGI) (www.itgi.org) se estableció en 1998 para promover el pensamiento internacional y los estándares relativos a la dirección y el control de las tecnologías de la información de una empresa. La efectividad en el gobierno de las tecnologías de la información (TI) permite asegurar a las empresas la consecución de sus objetivos, la optimización de las inversiones realizadas en TI y permite gestionar de forma adecuada los riesgos relacionados y las oportunidades ofrecidas por las mismas. El ITGI aporta investigación, recursos electrónicos y estudios de casos para ayudar a los líderes empresariales y los consejos de administración en las responsabilidades de gobierno de TI.

[11] La Information System Audit and Control Association (ISACA) es una asociación dedicada al gobierno de las tecnologías de la información. Es un miembro afiliado de la International Federation of Accountants (IFAC). Actualmente se hace referencia a la misma únicamente a través del acrónimo para reflejar la amplia gama de profesionales de tecnologías de información que acoge.

- Según [ONU77] una base de datos es una colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones.

- Según [FL05] una base de datos está constituida por el conjunto de datos de la empresa memorizado en un ordenador, que es utilizado por numerosas personas y cuya organización está regida por un modelo de datos.

- Según [DA88] se trata de un conjunto estructurado de datos registrados sobre soportes accesibles por ordenador para satisfacer simultáneamente a varios usuarios de forma selectiva y en tiempo oportuno.

- Según [Dee77] una base de datos está compuesta por una colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios.

- Según [Fra88] se puede decir que una base de datos está integrada por un conjunto de ficheros maestros, organizados y administrados de una manera flexible de modo que los ficheros puedan ser fácilmente adaptados a nuevas tareas imprevisibles.

- Según [EN08] una base de datos está constituida simplemente por un conjunto de datos interrelacionados.

La totalidad de definiciones recopiladas coinciden en que una base de datos es un conjunto, colección o depósito de datos almacenados en un soporte informático no volátil. Los datos están interrelacionados y estructurados de acuerdo con un modelo capaz de recoger el máximo contenido semántico. La base de datos se describe y se manipula apoyándose en dicho modelo.

Otra característica, que aparece concretamente en la primera definición mencionada, es que la redundancia de los datos debe ser controlada, de forma que no existan duplicidades perjudiciales ni innecesarias, y que las redundancias físicas, convenientes muchas veces a fin de responder a objetivos de eficiencia, sean tratadas por el mismo sistema, de modo que no puedan producirse inconsistencias.

Un aspecto importante que debe contemplar una definición completa de base de datos es el de independencia, tanto física como lógica, entre datos y tratamientos. Esta independencia, es una característica esencial que distingue las bases de datos de los sistemas de ficheros y que ha tenido especial influencia en la arquitectura de los sistemas gestores de base de datos.

La definición o descripción de los datos contenidos en la base de datos (lo que se denomina estructura o esquema de la base de datos) debe ser única y estar integrada con los mismos datos. En los sistemas basados en ficheros, los datos se encuentran almacenados en ficheros, mientras su descripción está separada de los mismos, habitualmente formando parte de los programas, para lo cual se precisa que los lenguajes faciliten medios para la descripción de los datos. En las bases de datos, la descripción y en algunos casos también una definición y documentación completas (metadatos), se almacena junto con los datos, de modo que estos datos están autodocumentados.

Finalmente, un elemento importante que debe considerar una definición completa es que la actualización y recuperación de los datos debe realizarse mediante procesos bien determinados, incluso en el sistema gestor de base de datos, el cual ha de proporcionar también instrumentos que faciliten el mantenimiento de la seguridad (confidencialidad, disponibilidad e integridad) del conjunto de datos.

Una definición completa y que contempla la totalidad de características enunciadas anteriormente, ha sido extraída de [MP99] y establece una base de datos como una colección o depósito de datos integrados, almacenados en un soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las abstracciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de datos.

3.2.2 Clasificación de BBDD atendiendo a la organización de la información

Una de las múltiples clasificaciones que se puede realizar sobre las bases de datos, atiende a los diferentes modelos de organización de la información que se pueden utilizar. Esta taxonomía se expone a continuación:

- Bases de datos jerárquicas: en este modelo, los datos se organizan de forma similar a un árbol, en donde un nodo padre de información puede tener varios hijos. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy distribuidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar de forma eficiente la redundancia de datos.

- Bases de datos en red: éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo (se permite que un mismo nodo tenga varios padres a diferencia del modelo jerárquico). Este modelo supuso una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que implica administrar la información en una base de datos de red, ha supuesto que sea un modelo que ha alcanzado poca difusión.

- Bases de datos relacionales: éste es el modelo actualmente más difundido. Su idea fundamental radica en el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". En la práctica, cada relación se representa como una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla). En este modelo, el lugar y la forma en la que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y potencia para administrar la información. El lenguaje más habitual para construir las consultas sobre bases de datos relacionales es SQL[12]. El sistema gestor de bases de datos Oracle constituye una implementación del modelo relacional.

- Bases de datos orientadas a objetos: las bases de datos orientadas a objetos (BDOO) son aquellas cuyo modelo de datos está orientado a objetos y almacenan y recuperan objetos en los que se almacena estado y comportamiento. Su origen se debe a que en los modelos clásicos de datos existen problemas para representar cierta información, puesto que aunque permiten representar gran cantidad de datos, las operaciones que se pueden realizar con ellos son bastante simples. Las clases utilizadas en un determinado lenguaje de programación orientado a objetos son las mismas clases que serán utilizadas en una BDOO; de tal manera, que no es necesaria una transformación del modelo de objetos para ser utilizado por un sistema gestor de bases de datos orientado a objetos (SGBDOO). De forma contraria, el modelo relacional requiere abstraerse lo suficiente como para adaptar los objetos del mundo real a tablas. Las bases de datos orientadas a objetos surgen para evitar los problemas que se originan al tratar de representar cierta información, aprovechar las ventajas del paradigma orientado a objetos en el campo de las bases de datos y para evitar transformaciones entre modelos de datos (usar el mismo modelo de objetos).

[12] SQL es acrónimo de Structured Query Language (Lenguaje de Consulta Estructurado), un lenguaje de acceso a bases de datos relacionales que permite la utilización de distintos tipos de operaciones sobre éstas. Gracias a la utilización del álgebra y de cálculos relacionales, SQL posibilita la realización de consultas para recuperar información de las bases de datos de forma sencilla.

- Bases de datos no relacionales (NOSQL) [Lop12]: estos términos hacen referencia a un subconjunto de bases de datos que difieren en varios aspectos de las bases de datos relacionales. Estas bases de datos no disponen de esquemas, no permiten el establecimiento de relaciones, no intentan garantizar las propiedades ACID[13] y son escalables de forma horizontal. A menudo ofrecen sólo garantías de consistencia débiles junto con la posibilidad de realizar transacciones sobre elementos de datos simples. Adicionalmente, emplean una arquitectura distribuida donde los datos se almacenan de forma redundante en distintos servidores, a menudo utilizando tablas Hash distribuidas. Suelen ofrecer estructuras de datos sencillas como Arrays asociativos o almacenes de pares clave-valor. Los principales tipos de bases de datos de acuerdo con su implementación son:
  - Almacenes de clave-valor.
  - Almacenes de familia de columnas.
  - Almacenes de documentos.
  - Almacenes de grafos.

Con respecto a esta taxonomía sería importante destacar el Teorema de CAP (Consistency, Availability, Partition tolerance) [GL12] que establece que en un sistema distribuido no es posible ofrecer de forma simultánea las tres propiedades que se exponen a continuación:

- Consistencia: todos los nodos del sistema distribuido tienen la capacidad de devolver la respuesta correcta a cada solicitud.

- Disponibilidad: en este caso la propiedad garantiza que cada solicitud debe recibir una respuesta.

- Tolerancia a la Partición: el sistema debe ser tolerante a la posible pérdida de comunicación de mensajes entre los nodos que lo componen.

De esta forma, y particularizando su aplicación sobre bases de datos relacionales y bases de datos no relacionales, obtendríamos el siguiente esquema que permite representar que características cumple cada una de estas clasificaciones:

Figura 2.Representación del Teorema de CAB extraída de [Lop12].

[13] ACID: acrónimo que hace referencia a las propiedades que deben cumplir las operaciones que se realizan dentro de una base de datos relacional: Atomicity (Atomicidad), Consistency (Consistencia), Isolation (Aislamiento) and Durability (Durabilidad).

3.2.3 Sistema gestor de base de datos (SGBD)

Los sistemas gestores de bases de datos (en inglés *database management system*, abreviado *dbms*) están compuestos por un tipo de software muy específico, dedicado a alojar bases de datos y a servir de interfaz entre las bases de datos, el usuario y las aplicaciones que las utilizan.

Una definición extraída de [MP99] describe el sistema gestor de base de datos como un conjunto coordinado de programas, procedimientos, lenguajes, etc. Este conjunto suministra a los distintos tipos de usuarios los medios necesarios para describir y manipular los datos almacenados en la base de datos, garantizando su seguridad.

El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos, que bajo una interpretación y un contexto se convertirán en información relevante para una organización.

Existen distintos objetivos que deben cumplir los sistemas gestores de bases de datos, en adelante sgbd:

- Abstracción de la información: los sgbd ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. No es relevante si una base de datos ocupa uno o cientos de archivos, puesto que este hecho se hace transparente, a priori, al usuario.

- Independencia: la independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que la utilizan.

- Consistencia: en aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario confirmar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma apropiada evitando la aparición de incoherencias.

- Seguridad: la información almacenada en una base de datos puede llegar a tener un gran valor. Los sgbd deben garantizar que esta información se encuentra segura y protegida, gracias a la definición de restricciones asociadas a los permisos de usuarios y de grupos de usuarios.

- Manejo de transacciones: una transacción es un proceso que se ejecuta de forma atómica, como una sola operación. Los sgbd proveen mecanismos para permitir las modificaciones sobre los datos implementando el concepto de transacción de forma que se mantenga la consistencia sobre los datos en todo momento.

- Tiempo de respuesta: el sgbd debe proporcionar un tiempo de respuesta adecuado tanto en las acciones de consulta de la información como en las acciones de actualización de los datos.

3.2.4 Ventajas de la utilización de un sgbd

Según

[PMM+05] las principales ventajas derivadas de la utilización de un sgbd son:

- Organización de la información a través de una estructura de datos común, accesible y reutilizable por diferentes usuarios y aplicaciones.

- Realizar un control centralizado de dicha información, lo que permite aumentar la integridad de los datos frente a caídas del sistema, prevenir ciertos tipos de inconsistencias, incrementar la fiabilidad y la disponibilidad de la información, eliminar la redundancia de los datos y establecer directivas de seguridad sobre el acceso a los datos.

- Mejorar el rendimiento del procesamiento de los datos mediante la utilización de estrategias de acceso y arquitecturas hardware/software optimizadas.

- La organización de la información conforme a los principios de independencia lógica y física reduce la necesidad de reescritura de programas frente a cambios estratégicos en los niveles inferiores. Además, la disposición de unos mecanismos de acceso a los datos perfectamente establecidos y formalizados, facilita el desarrollo de nuevas aplicaciones y la adaptación de las existentes a las nuevas necesidades de gestión.

- Se pueden priorizar las necesidades de acceso por parte de las aplicaciones adoptando diferentes criterios de asignación de recursos de procesamiento conforme a las necesidades.

- Gracias a la escalabilidad de algunos sistemas es posible incrementar la capacidad de procesamiento y el rendimiento del sgbd incorporando nuevos medios de almacenamiento, nuevos recursos de procesamiento (computadores, CPU´s, infraestructura de red…) sin que la percepción lógica de la base de datos por parte de las aplicaciones varíe.

- Organizan los datos con un impacto mínimo en el código de los programas.

- Disminuyen drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado si son bien explotados por los desarrolladores.

- Habitualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

3.3 Marco legal vigente

3.3.1 Introducción

Uno de los principales aspectos que ha propiciado la evolución de las técnicas de auditoría aplicables sobre los sgbd ha sido la aparición de normativa relativa a la protección de datos y en particular referente a los datos de carácter personal. Si bien, el conjunto de normas expuestas a continuación hacen referencia a la protección de datos personales, muchos de los aspectos mencionados pueden extenderse a información de distinta naturaleza, estableciendo criterios generales de obligado cumplimiento en materia de seguridad en los sgbd.

3.3.2 LOPD

3.3.2.1 Objeto y ámbito de aplicación

El texto legal que incorpora al ordenamiento jurídico español lo establecido por la Directiva 95/46/CE, es la Ley Orgánica 15/1999, de 13 de Diciembre, de Protección de Datos de carácter personal (LOPD) que entró en vigor el 14 de Enero de 2000.

El objeto de esta ley se encuentra definido en su artículo 1 y se concreta en "garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar".

El ámbito de aplicación de esta ley se define como "todos aquellos datos de carácter personal registrados en cualquier soporte que los haga susceptibles de tratamiento y cualquier modalidad de uso posterior de los mismos". La LOPD se aplica tanto a los tratamientos llevados a cabo por el sector público como por el sector privado.

3.3.2.2 Estructura de la Ley

La LOPD no presenta exposición de motivos y está dividida en 7 títulos que se enuncian a continuación:

- Título I Disposiciones generales: en este título se definen el objeto, ámbito de aplicación y aparecen las definiciones aplicables dentro del contexto de la propia ley.

- Título II Principios de la protección de datos: en este título se enuncian los aspectos fundamentales sobre los que se basa el tratamiento leal y transparente de los datos de carácter personal.

- Título III Derechos de las personas: en este título la ley establece un conjunto de derechos que toda persona puede ejercitar permitiendo a los afectados ejercer un papel activo en defensa de su privacidad.

- Título IV Disposiciones sectoriales:

    - Capítulo I: Ficheros de titularidad Pública: se definen como aquéllos cuya responsabilidad corresponde a las Administraciones Públicas (artículos 20 y 21), estableciendo ciertas especialidades en su régimen jurídico en las restantes disposiciones de este capítulo y en el artículo 46, en lo que se refiere al régimen sancionador.
    - Capítulo II: Ficheros de titularidad Privada: en líneas generales, aquéllos de los que son responsables empresas o personas físicas. También se consideran como tales aquellos ficheros cuyo responsable sea una entidad pública siempre que estos ficheros no estén vinculados al ejercicio de la función pública de dicha entidad.

- Título V Movimiento internacional de datos: permitido entre países con el mismo nivel o superior de seguridad, salvo autorización previa del Director de la Agencia. La principal excepción incluye el caso de la transferencia que tenga como destino un Estado miembro de la Unión Europea o respecto de la cual la Comisión Europea, en el ejercicio de sus competencias, haya declarado que garantiza un nivel de protección adecuado.

- Título VI Agencia de Protección de Datos: es un ente de derecho público con personalidad jurídica propia y plena capacidad pública y privada, que actúa con plena independencia de las AAPP[14] en el desarrollo de sus funciones. Entre las principales funciones se incluyen:

    - Velar por el cumplimiento de la legislación de protección de datos.
    - Atender las peticiones solicitadas por las personas afectadas.
    - Proporcionar información sobre los derechos de las personas.
    - Ejercer la potestad sancionadora.
    - Velar por la publicidad de la existencia de ficheros de carácter personal.
    - Redactar una memoria anual y remitida al Ministerio de Justicia.
    - Funciones de cooperación internacional.

- Título VII Infracciones y sanciones: las infracciones están divididas en leves, graves y muy graves.

A continuación, se exponen de forma más detallada tanto los principios de la protección de datos (Título II) como los derechos de las personas (Título III) como elementos susceptibles de considerarse durante un proceso de auditoría sobre una base de datos que albergue datos de carácter personal.

3.3.2.3 Principios de la protección de datos

Uno de los elementos fundamentales a considerar en el desarrollo de la Ley es la aparición del conjunto de principios de la protección de datos[15] que enuncian los aspectos fundamentales sobre los que se basa el tratamiento leal y transparente de los datos de carácter personal. En la exposición realizada a continuación se seguirá el orden y la forma de presentación que se adoptan en la LOPD:

- Calidad de los datos (Artículo 4 LOPD): marca las pautas que deberán seguirse en la recogida, almacenamiento y posterior utilización de los datos personales y, más concretamente los que se conocen como Principio de Proporcionalidad y Principio de Finalidad: sólo se deben recoger aquellos datos que sean adecuados, pertinentes y no excesivos en relación al ámbito y las finalidades para las que se hayan recogido; así mismo tampoco podrán utilizarse para fines incompatibles con aquellos para los que fueron recogidos. En cumplimiento del Principio de Exactitud, se establece la obligatoriedad de conservar los datos actualizados de tal forma que respondan con veracidad a la situación actual del afectado. Finalmente se dispone que los datos personales, no se conservarán, de forma personalizada, más allá del tiempo necesario de la finalidad para la que fueron recogidos.

[14] AAPP acrónimo de Administraciones Públicas.
[15] Recogidos en el Titulo II, artículos 4 a 12, de la LOPD.

- Derecho de la información en la recogida de los datos (Artículo 5 LOPD): define de forma clara e inequívoca la información que, obligatoriamente y con carácter previo, se debe comunicar al afectado al que se le requiera para suministrar cualquier dato personal. Dicha información constará de forma expresa, precisa e inequívoca de:

    - La existencia de un fichero o tratamiento de datos personales, de la finalidad de la recogida de datos personales y de los destinatarios de los mismos.
    - El carácter voluntario u obligatorio de las respuestas y de las consecuencias de la negativa a proporcionarlas.
    - La posibilidad del ejercicio de los derechos de acceso, rectificación y cancelación.
    - La identidad y dirección del responsable del tratamiento o, en su caso, de su representante.

  Si se utilizan cuestionarios o formularios, esta información deberá aparecer en los mismos de forma legible. Adicionalmente, se dispone que cuando los datos no se recaben del afectado, se deberá informar a éste en un plazo máximo de tres meses contados a partir del registro de los datos, de la existencia de un fichero o tratamiento de datos personales, de la finalidad de la recogida, de los destinatarios de la información, de la posibilidad de ejercitar los derechos de acceso, rectificación, cancelación y oposición y de la identidad y dirección del responsable del tratamiento o, en su caso, de su representante, salvo que el tratamiento tenga fines históricos, estadísticos o científicos o requiera esfuerzos desproporcionados a juicio de la Agencia de Protección de Datos u organismo autonómico equivalente. En el caso de comunicaciones comerciales que utilicen datos procedentes de fuentes accesibles al público, se informará al afectado en cada comunicación que se le realice, del origen de los datos, de la identidad del responsable del fichero y de los derechos que le asisten.

- Consentimiento del afectado (Artículo 6 LOPD): el consentimiento del afectado constituye la norma general de legitimación de los tratamientos de datos personales. Esto quiere decir, siguiendo la formulación de la LOPD, que "el tratamiento de los datos de carácter personal requerirá el consentimiento inequívoco del afectado, salvo que la ley disponga otra cosa". El consentimiento es revocable si existe causa justificada, pero la revocación no tiene efectos retroactivos respecto de tratamientos anteriores a la misma. No será necesario el consentimiento cuando se traten datos:

    - Los que recogen las Administraciones Públicas (AAPP) en el ejercicio de sus funciones.
    - Los que se refieren a las partes vinculadas por una relación de negocios, laboral, administrativa o un contrato o precontrato y sean necesarios para su mantenimiento o cumplimiento.
    - Cuando el tratamiento de los datos tenga por finalidad proteger un interés vital del interesado del artículo 7, apartado 6 de la LOPD.

- Que figuren en fuentes accesibles al público, pudiéndose considerar como tales sólo aquellas mencionadas en el artículo 3.j) de la LOPD, siempre que su tratamiento sea necesario para la satisfacción del interés legítimo perseguido por el responsable del fichero del tercero al que se comunican los datos, en tanto en cuanto no se vulneren los derechos y libertades fundamentales del interesado.

- Datos especialmente protegidos (Artículo 7 LOPD): la LOPD, en atención a la especial trascendencia que para la intimidad de las personas tiene el tratamiento de determinadas categorías de datos, configura, en su Artículo 7, un régimen de garantías especiales para aquellos datos que califica como "especialmente protegidos"[16]. En concreto, estas categorías son las siguientes:

  - Ideología, afiliación sindical, religión o creencias. Según establece la Constitución en su artículo 16.2, nadie podrá ser obligado a declarar sobre su ideología, religión o creencias. Este hecho, deberá, pues, ser comunicado al afectado con carácter previo a la recogida de datos. Adicionalmente, el tratamiento de las categorías de datos que encabezan este apartado requerirá el consentimiento expreso y por escrito en el caso de "ficheros mantenidos por partidos políticos, sindicatos, iglesias, confesiones o comunidades religiosas y asociaciones, fundaciones y otras entidades sin ánimo de lucro, cuya finalidad sea política, filosófica, religiosa o sindical, en cuanto a los datos relativos a sus asociados o miembros, sin perjuicio de que la cesión de dichos datos precisará siempre el previo conocimiento del afectado". Se excluye la necesidad del consentimiento expreso y por escrito para el tratamiento de los datos de ideología, afiliación sindical, religión o creencias en el caso de asociados o miembros de partidos políticos, sindicatos, iglesias, confesiones o comunidades religiosas y asociaciones, fundaciones y otras entidades sin ánimo de lucro, aunque dicho consentimiento siempre será necesario para ceder dichos datos.
  - Origen racial, salud y vida sexual. Sólo podrán recabarse, tratarse o cederse, cuando por razones de interés general así lo disponga una Ley o el afectado haya dado su consentimiento expreso.
  - Infracciones penales o administrativas: estos datos sólo podrán ser incluidos en los ficheros de las AAPP competentes y en los supuestos legalmente previstos.

[16] Estos tipos de datos son los que en la literatura de protección de datos se conocen como "datos sensibles" y su identificación como tales aparece ya en el artículo 6 del Convenio 108 del Consejo de Europa para la protección de personas con respecto al tratamiento automatizado de datos de carácter personal, habiéndose mantenido inalterable en todas las normas o directrices elaboradas desde entonces.

Adicionalmente, el apartado 6 de artículo 7 de la LOPD, establece como excepción, la necesidad del consentimiento del afectado en los tratamientos que se realicen con la finalidad de tratamiento, prevención o diagnóstico médico y la gestión de servicios sanitarios, siempre y cuando el tratamiento se realice por personal sanitario o sujeto a secreto profesional equivalente.

De la misma manera, se permite el tratamiento cuando sea necesario para salvaguardar el interés vital del afectado o de otra persona, cuando el afectado esté física o jurídicamente incapacitado para proporcionar el consentimiento o cuando la legislación sanitaria estatal autonómica así lo prevea.

- Seguridad de los datos (Artículo 9 LOPD): fija la obligación del responsable y del encargado del tratamiento de adoptar las medidas técnicas y organizativas necesarias para evitar la alteración, la pérdida, el tratamiento o el acceso no autorizado a los datos personales. Dichas medidas deberán tener en cuenta el estado de la tecnología, la naturaleza de los datos que deban protegerse y los riesgos a los que los mismos están sometidos.

- Deber de secreto (Artículo 10 LOPD): el responsable del fichero y todas las personas que intervengan en el tratamiento de datos personales (de todo tipo de datos personales y no exclusivamente protegidos) tienen la obligación de mantener el secreto profesional sobre los datos que conozcan, subsistiendo esta obligación aún después de haber finalizado su relación con el titular o el responsable del fichero.

- Comunicación o cesión de datos (Artículo 11 LOPD): al ser la comunicación de datos un tratamiento que entraña riesgos especiales para la intimidad de las personas, la LOPD, diseña un régimen de protección específico para el mismo. Los datos de carácter personal sólo podrán cederse en relación con fines legítimos del cedente y del cesionario (que por el mero hecho de serlo se obliga a observar lo establecido en la ley), siendo obligatorio el consentimiento del afectado. El consentimiento para la cesión es revocable. Existen excepciones a la obligatoriedad del consentimiento del afectado para la realización de cesiones cuando esté autorizado en una Ley:

    - Cuando se trate de datos procedentes de fuentes accesibles al público.
    - Cuando se derive de la libre aceptación de una relación jurídica cuyo desarrollo, cumplimiento y control implique necesariamente la realización de una cesión, que deberá limitarse a la finalidad que la justifique.
    - Cuando los destinatarios sean el Ministerio Fiscal o los Jueces y tribunales, el Defensor del Pueblo, el Tribunal de Cuentas (tampoco será necesario en el caso de cesiones a instituciones autonómicas con funciones análogas a las dos últimas citadas).
    - Cuando se produzca entre AAPP a los solos efectos de tratamiento de datos posterior con fines históricos, estadísticos o científicos.

Si los datos que se ceden han sido previamente disociados, no será de aplicación lo anteriormente expuesto, pues, en esencia, no existe cesión de datos personales.
Se considera nulo el consentimiento para la cesión si la información suministrada previamente al afectado no le permite conocer la finalidad para la que se ceden o el tipo de actividad del cesionario.

- Acceso a los datos por cuenta de terceros (Artículo 12 LOPD): bajo este epígrafe se regula la figura, funciones y obligaciones del encargado del tratamiento. En primer lugar, se establece que no se considera cesión el acceso a los datos cuando sea necesario para la prestación de un servicio al responsable del tratamiento. En segundo lugar se dispone que las condiciones bajo las que se realizan los trabajos, deberán formalizarse mediante un contrato, en el que se estipularán también las medidas de seguridad que el encargado del tratamiento está obligado a cumplir. Finalmente, la LOPD establece que si el encargado del tratamiento destina los datos a otra finalidad, los comunica a un tercero o incumple las estipulaciones del contrato, pasará a ser considerado como responsable del tratamiento y responderá a las infracciones en las que hubiera incurrido.

3.3.2.4 Derechos de las personas

Si los principios de protección de datos son, básicamente, los requisitos que todo aquel que quiera recoger y tratar datos personales debe cumplir para poder hacerlo legalmente, la Ley también atribuye a los afectados un papel activo en defensa de su privacidad. Para ello establece un conjunto de derechos que toda persona puede ejercer y que se enuncian a continuación:

- Impugnación de valoraciones (Artículo 13 LOPD): los afectados pueden no verse sometidos a una decisión con efectos jurídicos basada exclusivamente en un tratamiento de datos de datos destinados a evaluar su personalidad, y no solamente a impugnar el resultado de dicho tratamiento. Así mismo, aun en el caso de que el afectado se someta a un determinado tratamiento, éste tendrá derecho a obtener información sobre los criterios de valoración y el programa utilizados en el tratamiento que sirvió para adoptar una determinada decisión. Además, dicha valoración sólo podrá tener valor probatorio a petición del afectado.

- Derecho de consulta al Registro general de Protección de Datos (Artículo 14 LOPD): se establece el derecho a conocer, mediante consulta gratuita al Registro General de Protección de Datos (RGPD) de la Agencia de Protección de Datos, la existencia de aquellos ficheros que almacenan datos de carácter personal, la finalidad de dichos ficheros y la identidad del responsable de los mismos. La consagración de este derecho es la razón para la exigencia, de notificar al RGPD todos los tratamientos de datos personales que se lleven a cabo en los sectores público y privado.

- Derecho de acceso (Artículo 15 LOPD): los afectados tienen derecho a obtener información, de forma gratuita, sobre sus datos personales sometidos a tratamiento, el origen de dichos datos y las comunicaciones que se hayan realizado o esté previsto realizar. Este derecho se satisfará por parte de los responsables de ficheros utilizando cualquier medio conveniente, suministrando datos de forma clara y legible, sin utilizar claves o códigos. Es un derecho personalísimo[17] que sólo podrá ser ejercitado por el afectado ante el responsable del fichero, y, salvo interés legítimo, en intervalos no inferiores a doce meses. La petición de acceso se resolverá en un mes y la información suministrada al afectado deberá contener los datos de base del afectado, los datos resultantes de proceso informáticos, el origen de los datos, los cesionarios de los datos y los usos y finalidades para los que se almacenaron los datos[18]. Frente a la denegación del derecho de acceso, cabe reclamación ante el Director de la Agencia de Protección de Datos.

- Derechos de rectificación y cancelación (Artículo 16 LOPD): se instaura un plazo de diez días para hacer efectiva, de forma gratuita, la rectificación o cancelación solicitada por el afectado, implantándose, además, la obligación de rectificar o cancelar los datos no sólo cuando los mismos sean inexactos o incompletos, sino también cuando, genéricamente, su tratamiento no se ajuste a lo dispuesto en la LOPD. Si los datos rectificados o cancelados fueron cedidos, deberá modificarse la rectificación o cancelación de los mismos a los cesionarios de dichos datos. Asimismo, se explicita que la cancelación se ejecutará mediante el bloqueo de los datos, conservándose los mismos únicamente a disposición de los tribunales o de las Administraciones Públicas para la atención de las posibles responsabilidades nacidas del tratamiento. Una vez cumplido el plazo de prescripción de dichas responsabilidades, los datos serán suprimidos. No obstante, los datos de carácter personal se conservarán durante los plazos legalmente establecidos y según lo estipulado en relaciones contractuales. Contra la denegación de los derechos de rectificación o cancelación, cabe reclamación ante el Director de la Agencia de Protección de Datos en base a lo establecido en los apartados 1 y 2 del artículo 18 de la LOPD.

- Derecho de oposición[19] : adicionalmente, se consagra el derecho de oposición a un tratamiento específico por parte del afectado. Cuando el tratamiento no requiera el consentimiento del afectado y una Ley no disponga otra cosa, el afectado podrá oponerse a dicho tratamiento siempre y cuando existan motivos fundados y legítimos. Ante esta situación, el responsable del tratamiento excluirá del mismo los datos relativos al afectado.

[17] Artículo 11 del Real Decreto 1332/1994. También define como personalísimos los derechos de rectificación y cancelación.
[18] Artículo 12 del Real Decreto 1332/1994.
[19] Aunque este derecho se regula en el artículo 6.4 de la LOPD, se ha optado por incluirlo dentro del apartado dedicado a los derechos de las personas con la finalidad de alcanzar una mayor claridad expositiva.

- Tutela de los derechos (Artículo 18 LOPD): sí existen actuaciones contrarias a lo que establecen los principios de protección de datos o vulneraciones a los derechos de los ciudadanos, cabe la presentación de una reclamación ante la Agencia de Protección de Datos. La resolución del Director de la misma pone fin a la vía administrativa, cabiendo contra sus resoluciones recurso contencioso-administrativo. También asiste a los afectados el derecho a ser indemnizados si se les ha producido una lesión en sus bienes o derechos por actuaciones contrarias a la Ley. Dicha indemnización, deberá reclamarse ante los órganos de la jurisdicción ordinaria en el caso de ficheros privados y, según su legislación reguladora, en el caso de ficheros públicos.

3.3.2.5 Real Decreto 1720/2007

El Real Decreto 1720/2007 nace con la vocación de desarrollar, no sólo los mandatos contenidos en la LOPD de acuerdo con los principios que emanan de la Directiva sino también, aquellos que en los años de vigencia de la LOPD se ha demostrado que precisan de un mayor desarrollo normativo. En particular, nos centraremos en el Título VIII que establece las medidas de seguridad en el tratamiento de datos de carácter personal, y más concretamente en el Capítulo III en el que se destacan las medidas de seguridad aplicables a ficheros y tratamientos automatizados. Los siguientes puntos se centran en los artículos que tienen especial relación con aquellas medidas de seguridad que pueden adoptarse a través de la utilización de un sgbd, particularizando su implementación en el sgbdr de Oracle.

3.3.2.5.1 Artículo 91. Control de acceso

En este artículo se destaca la necesidad de que los usuarios tengan acceso únicamente a aquellos recursos que precisen para el desarrollo de sus funciones. Adicionalmente destaca que el responsable del fichero establecerá mecanismos para evitar que un usuario pueda acceder a recursos con derechos distintos de los autorizados.

Implementación en un sgbdr

Particularizando la implementación de esta necesidad en el sgbdr de Oracle, se puede destacar el sistema de roles y privilegios[20]. A través de la utilización de este sistema se establecen restricciones de acceso sobre determinada información permitiendo su utilización, únicamente a usuarios autorizados.

3.3.2.5.2 Artículo 93. Identificación y autorización

A través de este artículo se establece la obligatoriedad de adoptar las medidas que garanticen la correcta identificación y autenticación de los usuarios.

[20] Encontrará una explicación más detallada del sistema de roles y privilegios en el punto "3.4.1.1.2 Implementación en un sgbdr".

Adicionalmente destaca que en el caso de que, cuando el mecanismo de autenticación esté basado en contraseñas, existirá un procedimiento de asignación, distribución y almacenamiento que garantice su confidencialidad e integridad. En el documento de seguridad[21] se establecerá la periodicidad, que en ningún caso será superior a un año, con la que tienen que ser cambiadas las contraseñas que, mientras estén vigentes, se almacenarán de forma ininteligible.

Implementación en un sgbdr

Dentro del sgbdr de Oracle la seguridad del sistema se gestiona y controla a través del concepto de usuario de base de datos. Las contraseñas son el método más común con el que Oracle permite la autentificación cuando se entra en una base de datos.

Cuando se accede a una base de datos [Ben09] se necesita introducir una password para compararla con algo que ya esté almacenado previamente. Cuando a través de un cliente se establece una conexión con Oracle, se necesita analizar la password introducida y enviar algo al servidor de base de datos. El sevidor de base de datos compara aquello que se ha enviado con la propia password que está almacenada en el servidor. Las password en Oracle nunca son enviadas sin cifrar y tampoco son almacenadas sin cifrar. En Oracle 11g las password son cifradas utilizando Adavanced Encription Standard (AES) antes de enviarse a través de la red. La forma de trabajar del sgbdr Oracle se describe a continuación:

- El cliente envía el nombre de login que se está utilizando a la base de datos.
- La base de datos envía una comprobación creada a través del cifrado de un número aleatorio que contiene el Hash[22] asociado al password de usuario. La clave aleatoria puede ser vista como una clave de sesión y por tanto es diferente para cada una de las sesiones.
- El cliente procesa el Hash asociado al password y descifra el número aleatorio. En este instante, el cliente y el servidor utilizan una clave de sesión común.
- El cliente utiliza este número aleatorio como clave para cifrar la password y envarla al servidor de base de datos.

Cuando las password son almacenadas en USER$[23], no son almacenadas sin cifrar. En lugar de un valor sin cifrar se almacena el resultado de aplicar una función Hash de un único sentido.

[21] El documento de seguridad, descrito en el Capítulo II del presente Real Decreto, recogerá las medidas de índole técnica y organizativa acordes a la normativa de seguridad vigente, que será de obligado cumplimiento para el personal con acceso a los sistemas de información.
[22] Las funciones Hash son funciones computables mediante un algoritmo. La función Hash [Ben09] en este caso, está basada en el en el Secure Hash Algorithm-1 (SHA-1) considerado una funciones de un único sentido utilizables en la actualidad.
[23] Tabla perteneciente al esquema SYS que almacena información relevante de los usuarios.

Siempre que se crea un usuario [Ben09], se le asigna un perfil por defecto de gestión de password. Este tipo de perfil permite especificar cuántos intentos fallidos de entrada se permiten antes de bloquear la cuenta, qué tipo de passwords se aceptan (en términos de complejidad de la password) y cuánto tiempo se permite mantener una password antes de que el sgbdr fuerce a cambiarla. Constituye el trabajo de un Administrador asegurarse de que la totalidad de usuarios disponen un perfil de password adecuado. A continuación podemos ver un ejemplo de establecimiento de perfil de password:

```
SQL>create profile ejemplo limit
  2    failed_login_attempts 5
  3    password_life_time   60
  4    password_reuse_time 60;
```

Figura 3. Ejemplo de creación de un perfil de password.

Lo que se ha definido en la figura anterior es un perfil en el que una cuenta de usuario que lo utilice será bloqueada si hay cinco intentos fallidos de login, el password expirará tras 60 días y que el mismo password no puede ser reutilizado en un periodo de 60 días. Es posible cambiar los atributos de un perfil de password de la siguiente forma:

```
SQL> alter profile ejemplo limit password_reuse_max 10;
```

Figura 4. Ejemplo de modificación de un perfil de password.

Cuando se procede a crear un usuario se asigna el perfil de password para establecer las restricciones relativas a la password asociadas a dicho usuario:

```
SQL> create user usuario1 identified by <pwd>;
SQL> grant create session to usuario1;
SQL> alter user usuario1 profile ejemplo;
```

Figura 5. Ejemplo de creación de un usuario, concesión de privilegio y asignación de perfil de password.

La lista completa de atributos que se pueden utilizar para establecer un perfil de password se muestra a continuación. En Oracle 11g, si se omite algún parámetro, asume un valor por defecto que también aparece reflejado en la siguiente tabla:

Atributo	Descripción	Valor por defecto
FAILED_LOGIN_ATTEMPTS	Número de intentos fallidos de entrada, antes de que la cuenta de usuario sea bloqueada.	10
PASWORD_LIFE_TIME	Número de días después de los que una password debe ser cambiada antes de convertirse en no válida.	180
PASSWORD_REUSE_TIME	Intervalo de días dentro del cual la misma password no puede ser reutilizada.	No
PASSWORD_LOCK_TIME	Días que deben pasar después de que una cuenta es bloqueada o antes de que es desbloqueada.	1
PASSWORD_GRACE_TIME	Cuando una password expira este parámetro permite añadir un periodo adicional durante el que un usuario puede utilizar la password antigua.	7
PASSWORD_VERIFY_FUNCTION	Permite especificar una función para validar la fortaleza de una password.	No

Tabla 2. Parámetros utilizables en la asignación de un perfil de password extraída de [Ben09].

3.3.2.5.3 Artículo 94. Copias de respaldo y recuperación

En este artículo se establece la obligatoriedad de dotar de procedimientos de actuación para la realización de copias de seguridad. Adicionalmente, se establecerán procedimientos para la recuperación de los datos que garanticen en todo momento su reconstrucción en el estado en el que se encontraban en el instante en el que se produjo la pérdida o destrucción.

Implementación en un sgbdr

Los sgbd suministran herramientas que permiten llevar a cabo las acciones de backup y restauración sobre una base de datos. Particularizando estas operaciones sobre el sgbdr de Oracle podemos encontrar diversas herramientas que contemplan y facilitan esta labor como: Data Pump Export, modo Archivelog y Recovery Manager (RMAN)[24].

3.3.2.5.4 Artículo 96. Auditoría

En este artículo se establece la necesidad de realización de auditorías internas o externas, al menos cada dos años, con la finalidad de verificar el cumplimiento del Título VIII. El informe de auditoría resultante deberá dictaminar sobre la adecuación de las medidas y controles a la Ley y su desarrollo reglamentario, identificar sus deficiencias y proponer medidas correctoras o complementarias necesarias. Deberá, igualmente, incluir los datos, hechos y observaciones en que se basen los dictámenes alcanzados y las recomendaciones propuestas.

[24] Encontrará una definición más detallada de estas herramientas en el punto "3.4.1.4 Implementación en un sgbdr".

Implementación en un sgbdr

Existen determinadas herramientas[25] que permiten la generación de registros de auditoría que ayudan a determinar el autor, la acción, la naturaleza del cambio realizado, el instante en el que se produjo y los medios que se utilizaron. Dentro del sgbdr de Oracle existen varias posibilidades cómo la utilización de la denominada Auditoría Estándar, la Auditoría de Grano Fino y la utilización de la herramienta Oracle Audit Vault[26].

3.3.2.5.5 Artículo 98. Identificación y autenticación

En el Artículo 98 se establece la obligatoriedad de implantar un mecanismo que limite la posibilidad de intentar reiteradamente el acceso no autorizado al sistema de información.

Implementación en un sgbdr

En el punto "3.3.2.5.2 Artículo 93. Identificación y autorización" se han comentado los mecanismos aplicables en la definición de usuarios para habilitar la utilización de perfiles de password. Dentro de la utilización de perfiles de password, se ha mencionado el parámetro *FAILED_LOGIN_ATTEMPS* que se utiliza, particularmente, con la finalidad de limitar el número de intentos seguidos de conexión bloqueando la cuenta de usuario sí se supera el límite establecido por dicho parámetro.

3.3.2.5.6 Artículo 101. Gestión y distribución de soportes

En este artículo se establece la necesidad de cifrar los datos en la distribución de soportes que contengan datos de carácter personal o bien utilizando otro mecanismo que garantice que dicha información no sea accesible o manipulada durante su transporte. Así mismo, se cifrarán los datos que contengan los dispositivos portátiles cuando estos se encuentren fuera de las instalaciones.

Implementación en un sgbdr

Sobre la mayoría de sgbdr es posible utilizar distintos algoritmos de cifrado que permiten proteger la información contenida en los mismos. Los algoritmos de cifrado de datos que el sgbdr de Oracle soporta son los siguientes:

- Algoritmos DES: algoritmo de cifrado de clave simétrica.
- Triple DES: algoritmo más seguro que DES puesto que aplica un triple cifrado DES.
- RSA: algoritmo de cifrado de clave pública o asimétrica.

[25] La herramienta AAS11 desarrollada dentro del ámbito de este proyecto puede constituir un elemento de apoyo más en la realización de este tipo de auditorías con la finalidad de ayudar a detectar debilidades y posibles puntos de mejora.
[26] Encontrará una descripción de estas herramientas en el punto "3.4.1.4 Implementación en un sgbdr".

3.3.2.5.7 Artículo 103. Registro de accesos

En este artículo se destaca la obligatoriedad de guardar determinada información sobre los intentos de accesos que se produzcan. La información que se registrará por cada intento de acceso, como mínimo, deberá incluir: la identificación del usuario, la fecha y hora en la que se realizó, el fichero accedido, el tipo de acceso y si ha sido autorizado o denegado. En el caso de que el acceso haya sido autorizado, será preciso guardar la información que permite identificar el registro accedido.

Implementación en un sgbdr

Particularizando su aplicación dentro del sgbdr de Oracle, se dispone de la posibilidad de utilizar la Auditoría Estándar [Ben09] que permite auditar actividades basándose en el tipo de actividad, objeto, privilegio o usuario. Esta herramienta puede ser configurada para registrar los accesos a determinados elementos.

3.3.2.5.8 Artículo 104. Telecomunicaciones

Este artículo establece que la transmisión de datos de carácter personal a través de redes públicas o redes inalámbricas de comunicaciones electrónicas se realizará cifrando dichos datos o bien utilizando cualquier otro mecanismo que garantice que la información no sea inteligible ni manipulada por terceros.

Implementación en un sgbdr

Algunos sgbdr suministran herramientas que permiten garantizar la seguridad en un entorno de red. El sgbdr Oracle ofrece la herramienta Oracle Advanced Security[27].

3.4 Auditoría sobre un sistema gestor de base de datos

En este punto se establecen el conjunto de principios de auditoría aplicables sobre un sgbd. Para extraer estos principios se ha tomado como base tanto el estándar internacional ISO/IEC 27002 [ISO/IEC05] como el COBIT 4.1[28] [ITGI07]. En los siguientes apartados se detallarán los principios extraídos de esta documentación y su relación con un proceso de auditoría sobre un sgbdr.

[27] Encontrará un descripción de la herramienta Oracle Advanced Security en el punto "3.4.1.5.2 Implementación en un sgbdr".
[28] La versión 5 de este marco de trabajo está disponible desde Junio de 2012.

3.4.1 Estándar Internacional ISO/IEC 27002

El estándar internacional ISO/IEC 27002 constituye una referencia sobre la práctica de la gestión de la seguridad de la información aplicable en entornos tecnológicos. Este documento proporciona recomendaciones sobre buenas prácticas relativas a la gestión de la seguridad de la información. Cada uno de los puntos que se mencionan a continuación identifica un riesgo o un requerimiento considerado como crítico. Por cada uno de estos requerimientos o riesgos se define un control para permitir su detección y se proponen un conjunto de directrices para su implementación. Finalmente, se propone información adicional asociada a dicho control o a su implementación. A continuación, se detallarán aquellos puntos que tienen particular relación con los sgbdr.

3.4.1.1 Clasificación de la información

En el punto 7.2[29] de este estándar se expone la necesidad de clasificar la información para permitir establecer prioridades, grados de protección sobre la misma e identificar requerimientos asociados.

La información tiene diversos grados de confidencialidad e importancia. Determinados elementos pueden requerir un nivel de protección adicional o manejo especial. Debe utilizarse un esquema de clasificación de información para definir un conjunto apropiado de niveles de protección y comunicar la necesidad de medidas de uso especiales.

3.4.1.1.1 Directrices de clasificación

Control

Debería realizarse una clasificación de la información en términos de su valor, requerimientos legales, sensibilidad y grado crítico para la organización.

Directrices de implementación

Las clasificaciones y los controles de protección asociados para la información deberían tomar en cuenta las necesidades comerciales de intercambiar o restringir información y los impactos comerciales asociados con dichas necesidades.

Las directrices de clasificación deberían incluir protocolos para la clasificación inicial y la reclasificación a lo largo del tiempo; en concordancia con alguna política pre-determinada de control de acceso.

Debería ser responsabilidad del propietario de un determinado activo definir la clasificación de dicho activo, revisarla periódicamente y asegurarse que se mantenga actualizada y en el nivel apropiado.

[29] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] página 21.

Se debería tener en consideración el número de categorías de clasificación y los beneficios a obtenerse con su uso. Los esquemas demasiado complejos pueden volverse difíciles de utilizar o poco prácticos.

Otra información

Se puede evaluar el nivel de protección analizando la confidencialidad, integridad y disponibilidad, y cualquier otro requerimiento para la información considerada.

Con frecuencia, la información deja de ser sensible o crítica después de un cierto período de tiempo, por ejemplo, cuando la información se ha hecho pública. Se deberían tomar en cuenta estos aspectos, ya que la "sobre-clasificación" puede llevar a la implementación de controles innecesarios, resultando en un gasto adicional.

Agrupar documentos con requerimientos de seguridad similares cuando se asignan niveles de clasificación podría ayudar a simplificar la tarea de clasificación.

En general, la clasificación dada a la información es una manera rápida para determinar cómo se está manejando y protegiendo la información.

3.4.1.1.2 Implementación en un sgbdr

Un sgbdr dota de los mecanismos necesarios para permitir llevar a cabo una clasificación de la información que además de aportar organización permita restringir el acceso a personal no autorizado. Estos mecanismos se describen a continuación:

- Esquema de base de datos: describe la estructura de una base de datos, en un lenguaje formal soportado por un sgbdr.

- Tabla: elemento componente del esquema que agrupa un conjunto de tuplas relacionadas. La información contenida en estas tuplas podrá descomponerse en una serie de campos.

- Relación: Interconexión que se establece entre tablas a través de un conjunto de campos.

- Sistema de roles y privilegios: La mayoría de sgbdr están dotados de sistemas de asignación de roles y privilegios a los usuarios de tal forma que permiten establecer restricciones de acceso sobre determinada información. Este tipo de mecanismos permiten establecer grados de protección sobre la información permitiendo únicamente acceso a la misma a usuarios autorizados. Durante la etapa de diseño de una base de datos se deben tener en cuenta la totalidad de estos aspectos con el objetivo de cumplir los requisitos preestablecidos.

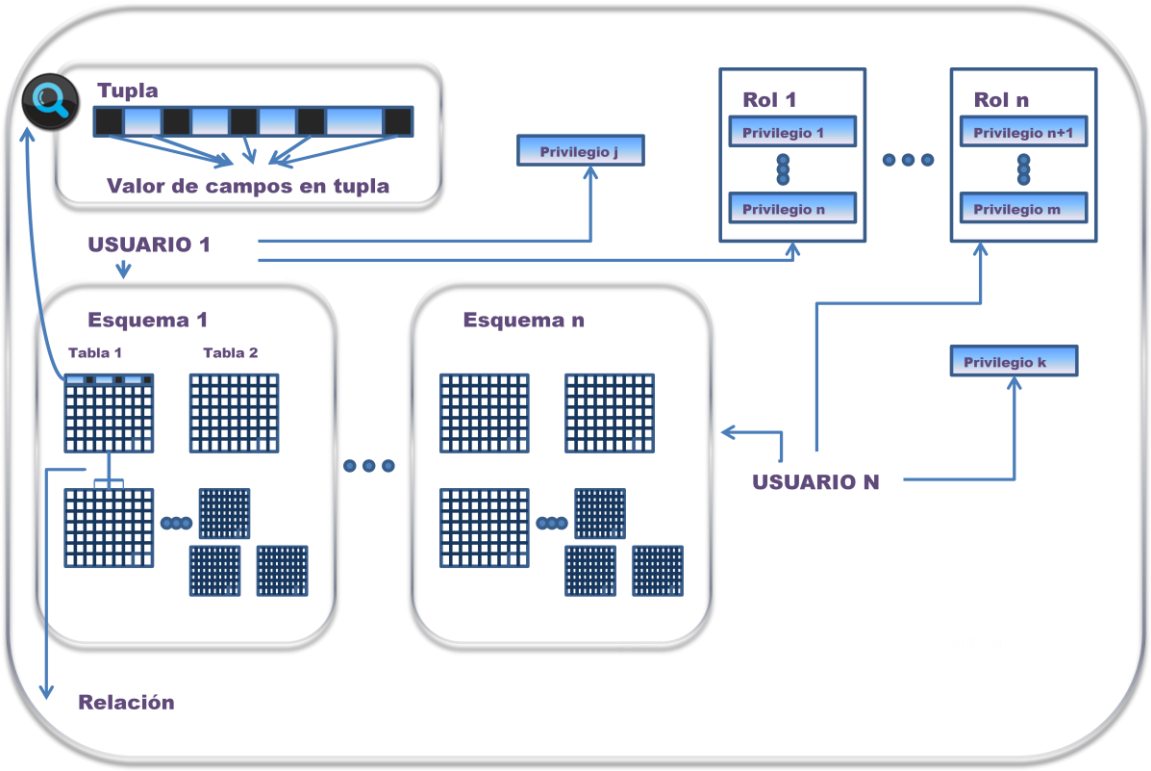


Figura 6. Relación entre conceptos de esquema, tabla, tupla, columna, usuarios, roles y privilegios.

3.4.1.2 Condiciones previas a la contratación

En el apartado 8.1[30] del estándar se puntualizan una serie de condiciones previas que permiten garantizar que empleados[31], contratistas y terceros entienden sus responsabilidades, y sean idóneos para los roles para los cuales son considerados; y reducir el riesgo de robo, fraude y mal uso de los medios.

Las responsabilidades de seguridad deberían ser tratadas antes del empleo en descripciones de trabajo adecuadas y en los términos y condiciones del empleo.

Este apartado, a pesar de contemplar una serie de condiciones muy generales, también menciona tanto la definición de roles y responsabilidades como el retiro de los derechos de acceso. A continuación se expondrán cada uno de estos puntos y se particularizará su aplicación sobre el ámbito de los sgbdr.

[30] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en páginas 23-24.

[31] Dentro del ámbito de este apartado la palabra "empleo" se utiliza para abarcar las siguientes situaciones diversas: empleo de personas (temporal o permanente), asignación de roles de trabajo, asignación de contratos y la terminación de cualquiera de estos acuerdos.

3.4.1.2.1 Roles y responsabilidades

Control

Se deberían definir y documentar los roles y responsabilidades de la seguridad de los empleados, contratistas y terceros en concordancia con la política de seguridad de la información de la organización.

Directrices de implementación

Los roles y responsabilidades deberían incluir requerimientos para:

- Implementar y actuar en concordancia con las políticas de seguridad de la información de la organización.
- Proteger los activos contra el acceso, divulgación, modificación, destrucción o interferencia no autorizada.
- Ejecutar procesos o actividades de seguridad particulares.
- Asegurar que se asigne a la persona la responsabilidad por las acciones tomadas.
- Reportar eventos de seguridad o eventos potenciales u otros riesgos de seguridad para la organización.
- Los roles y responsabilidades de la seguridad deberían ser definidos y claramente comunicados a los candidatos para el puesto durante el proceso de pre-empleo.

Otra información

Se pueden utilizar las descripciones del puesto para documentar los roles y responsabilidades de seguridad. También se deberían definir y comunicar claramente los roles y responsabilidades para las personas no contratadas directamente a través de la organización; por ejemplo, a través de una tercera organización.

3.4.1.2.2 Implementación en un sgbdr

Particularizando el ámbito de aplicación de este apartado dentro de la información que reside en un sgbdr existe un sistema de privilegios y roles que permite limitar y restringir el acceso a la información. Para describir el sistema de privilegios utilizado en Oracle se aplicarán las siguientes definiciones [Ben09] :

- Privilegio: permiso para realizar una determinada acción, asignable directamente a un usuario o un rol.

- Rol: agrupación de privilegios bajo un identificador, asignable a usuarios o a roles.

- Usuario: agrupación de objetos y privilegios, accesibles a través de un identificador y una palabra clave.

- Perfil: conjunto de restricciones relativas al uso de recursos, y asignable a usuarios. Un usuario puede estar únicamente asignado a un perfil.

- Recurso: elemento sobre el que es posible aplicar una restricción para permitir su asignación a un perfil.

La relación entre estos conceptos se expone en la siguiente figura:

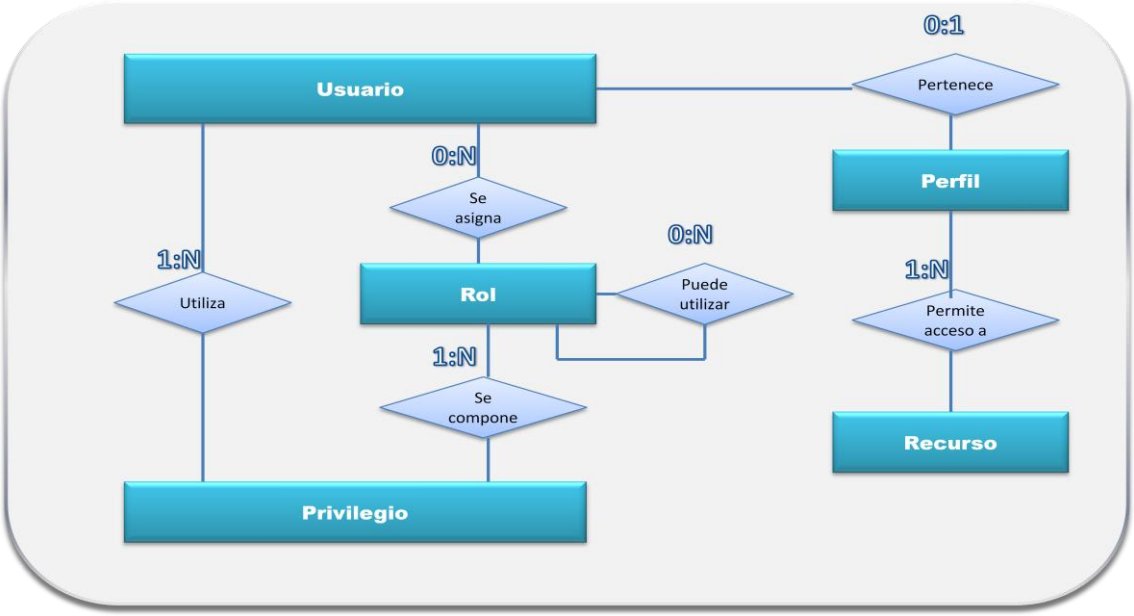


Figura 7. Relación entre conceptos de usuario, rol, privilegio, perfil y recurso.

En la figura expuesta se puede observar el modelo de privilegios utilizado en el sgbdr de Oracle a través del cual se puede permitir o denegar acceso a los elementos de una base de datos. Como se puede ver, un determinado usuario puede disponer de una serie de privilegios. Los privilegios, a su vez, pueden estar asignados a una serie de roles que facilitan su gestión. Estos roles pueden componerse, además de privilegios, de otros roles permitiendo construir una jerarquía. Finalmente, un usuario puede pertenecer a un determinado perfil que habilitará el acceso a los recursos que estén asociados al mismo.

3.4.1.3 Terminación o cambio de empleo

En el apartado 8.3[32] del estándar se destacan el conjunto de condiciones que permiten asegurar que los usuarios empleados, contratistas y terceras personas salgan de la organización o cambien de empleo de una manera ordenada.

Se deberían establecer las responsabilidades para asegurar que la salida de la organización del usuario empleado, contratista o tercera persona sea manejada y se complete la devolución de todo el equipo y se eliminen todos los derechos de acceso.

Los cambios en las responsabilidades y empleos dentro de la organización se pueden manejar como la terminación de la responsabilidad o empleo respectivo en concordancia con esta sección.

Las condiciones mencionadas en este apartado se particularizarán para su aplicación dentro de un sgbdr.

3.4.1.3.1 Retiro de los derechos de acceso

Control

Los derechos de acceso de todos los usuarios empleados, contratistas y terceras personas a la información y los medios de procesamiento de información deberían ser retirados a la terminación de su empleo, contrato o acuerdo, o en cualquier caso, deberían ser reajustados de acuerdo al cambio.

Directrices de implementación

A la terminación, se deberían reconsiderar los derechos de acceso de una persona a los activos asociados con los sistemas y servicios de información. Esto determinará si es necesario retirar los derechos de acceso. Los cambios de empleo se deberían reflejar en el retiro de todos los derechos de acceso que no han sido aprobados para el nuevo empleo. Los derechos de acceso que se deberían retirar o adaptar incluyen el acceso físico y lógico, llaves, tarjetas de identificación, medios de procesamiento de información, suscripciones; y el retiro de cualquier documentación que identifique a la persona como miembro actual de la organización. Si un usuario empleado, contratista o tercera persona que está dejando la organización conoce las claves secretas para las cuentas aún activas, éstas deberían ser cambiadas a la terminación o cambio del empleo, contrato o acuerdo.

Los derechos de acceso para los activos de información y los medios de procesamiento de información se deberían reducir o retirar antes de la terminación o cambio del empleo, dependiendo de la evaluación de los factores de riesgo como:

[32] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en páginas 27-28.

- Si la terminación o cambio es iniciado por el usuario empleado, contratista o tercera persona, o por la gerencia y la razón de la terminación.

- Las responsabilidades actuales del usuario empleado, contratista o cualquier otro usuario.

- El valor de los activos actualmente disponibles.

Otra información

En ciertas circunstancias, los derechos de acceso pueden ser asignados con la finalidad de estar disponibles para más personas que el usuario empleado, contratista o tercera persona; por ejemplo, los ID´s del grupo. En tales circunstancias, las personas que se van deberían ser retiradas de las listas de acceso del grupo y se deberían instaurar procesos para comunicar a todo el resto de usuarios empleados, contratistas y terceros involucrados para que ya no compartan esta información con la persona que abandona la organización.

En casos de terminaciones iniciadas por la gerencia, los empleados, contratistas o terceros descontentos pueden corromper la información deliberadamente o sabotear los medios de procesamiento de la información. Las personas que renuncian, podrían tratar de recolectar información para su uso futuro.

3.4.1.3.2 Implementación en un sgbdr

El procedimiento de retirada de permisos a un usuario en un sgbdr o incluso la baja del mismo, debe de ser llevada a cabo por el propio Administrador de base de datos. El Administrador de base de datos dispondrá de la potestad para limitar el acceso a un determinado usuario, conceder privilegios a otros usuarios sobre determinados elementos o incluso llevar a cabo una eliminación completa de un esquema asociado a un usuario.

3.4.1.4 Respaldo o backup

En el punto 10.5[33] del estándar se plantea la necesidad mantener la integridad y disponibilidad de la información y los medios de procesamiento de información.

Se deberían establecer los procedimientos de rutina para implementar la política de respaldo acordada y la estrategia para llevar a cabo copias de seguridad de los datos y practicar su restauración oportuna.

Control

Se debería hacer copias de seguridad de la información y software y se deberían probar regularmente, en concordancia con la política de copias de seguridad establecida.

[33] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en página 44.

Directrices de implementación

Se deberían proporcionar medios de respaldo adecuados para asegurar que toda la información esencial y software se pueda recuperar después de un desastre o fallo en los medios de almacenamiento. Se deberían considerar los siguientes elementos para el respaldo de la información:

- Se debería definir el nivel necesario de respaldo de la información.
- Se deberían producir registros exactos y completos de las copias de seguridad y procedimientos documentados de la restauración.
- La extensión (por ejemplo, copia de seguridad completa o parcial) y la frecuencia de las copias de seguridad, debería reflejar los requerimientos comerciales de la organización, los requerimientos de seguridad de la información involucrada, y el grado crítico de la información para garantizar la operación continuada de la organización.
- Las copias de seguridad se deberían almacenar en un lugar apartado, a la distancia suficiente como para escapar de cualquier daño por un desastre en el local principal.
- A la información de respaldo se le debería dar el nivel de protección física y ambiental apropiado consistente con los estándares aplicados en el local principal. Los controles aplicados a los medios de almacenamiento en el local principal se deberían extender para cubrir la ubicación de la copia de respaldo.
- Los medios de respaldo se deberían probar regularmente para asegurar que se pueda confiar en ellos para usarlos cuando sea necesario, en caso de emergencia.
- Los procedimientos de restauración se deberían chequear y probar regularmente para asegurar que sean efectivos y que pueden ser completados dentro del tiempo asignado en los procedimientos operacionales establecidos por la organización para la recuperación.
- En situaciones en las que la confidencialidad es de importancia, las copias de respaldo deberían ser protegidas por medio de una codificación. Los procedimientos de respaldo para los sistemas individuales deberían ser probados regularmente para asegurar que cumplan con los requerimientos de los planes de continuidad establecidos por la organización. Para sistemas críticos, los procedimientos de respaldo deberían abarcar toda la información, aplicaciones y datos de todos los sistemas, necesarios para recuperar el sistema completo en caso de un desastre. Se debería determinar el período de retención para la información comercial esencial, y también cualquier requerimiento para que las copias de archivo se mantengan permanentemente.

Otra información

Los procedimientos de respaldo pueden ser automatizados para facilitar el proceso de respaldo y restauración. Estas soluciones automatizadas deberían ser probadas suficientemente antes de su implementación y revisadas en intervalos regulares para comprobar su exactitud.

3.4.1.4.1 Implementación en un sgbdr

Un sgbdr suministra herramientas que permiten llevar a cabo las acciones de backup y restauración sobre una base de datos. Particularizando la aplicación de las acciones de backup en el sgbdr de Oracle existen diferentes herramientas:

- Data Pump Export [Her11]: es una utilidad de Oracle que se encarga de copiar los datos a un fichero resultado con extensión ***.dmp*** en la base de datos ORACLE.

- Modo Archivelog[34]: permite que el sgbdr de Oracle almacene el grupo de ficheros de actualización antes de que el proceso LGWR realice nuevas escrituras sobre el mismo. Si la base de datos está configurada en modo Archivelog pueden realizarse backups mientras la base de datos está en funcionamiento. Este hecho permitirá realizar una restauración de la base de datos sin pérdida de datos.

- Recovery Manager (RMAN) [Heu09]: RMAN es una herramienta de línea de comandos que permite realizar copias de seguridad y recuperaciones de una base de datos. Esta herramienta utiliza un repositorio para almacenar información de su configuración, las copias de seguridad realizadas, la estructura de la base de datos destino, los ficheros de actualización almacenados…Este repositorio se almacena en el fichero de control de la base de datos destino. El propio repositorio puede almacenarse en un catálogo de recuperación que se implementa por medio de un esquema en otra base de datos. Un solo catálogo de recuperación se puede utilizar para centralizar los repositorios RMAN de varias bases de datos objetivo.

3.4.1.5 Gestión de la seguridad de la red

En el punto 10.6[35] del estándar se expone la necesidad de asegurar la protección de la información en redes y la protección de la infraestructura de soporte.

La gestión segura de las redes, la cual puede abarcar los límites organizacionales, requiere de la cuidadosa consideración del flujo de datos, implicaciones legales, monitorización y protección.

También se pueden requerir controles adicionales para proteger la información confidencial que pasa a través de redes públicas.

[34] En el Capítulo 4: El Sistema Gestor de Base de Datos Oracle 11g se realiza una exposición detallada de la arquitectura que se emplea en este sistema, que facilitará la comprensión de este modo de funcionamiento.
[35] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en página 45.

3.4.1.5.1 Controles de redes

Control

Las redes deberían ser adecuadamente manejadas y controladas para poder proteger la información que circula a través de las mismas para así preservar la seguridad de los sistemas y aplicaciones que las utilizan.

Directrices de implementación

Los administradores de la red deberían implementar controles para garantizar la seguridad de la información en las redes, y proteger los servicios conectados de accesos no autorizados. En particular, se deberían considerar los siguientes elementos:

- Cuando sea apropiado, la responsabilidad operacional para las redes se debería separar de las operaciones de cómputo.

- Se deberían establecer las responsabilidades y procedimientos para la gestión de los equipos remotos, incluyendo los equipos ubicados en las áreas del usuario.

- Se deberían establecer controles especiales para salvaguardar la confidencialidad y la integridad de los datos que pasan a través de las redes públicas o a través de las redes inalámbricas y controlar los sistemas y aplicaciones conectados. También se pueden requerir controles especiales para mantener la disponibilidad de los servicios de la red y los ordenadores conectados.

- Se deberían aplicar registros de ingreso y monitorización apropiados para permitir el registro de las acciones de seguridad relevantes.

- Las actividades de gestión deberían estar estrechamente coordinadas para optimizar el servicio a la organización y para asegurar que los controles sean aplicados consistentemente a través de la infraestructura de procesamiento de la información.

Otra información

Se puede encontrar información adicional sobre la seguridad de la red en ISO/IEC 18028 [ISO/IEC06].

3.4.1.5.2 Implementación en un sgbdr

Con respecto a la incorporación de mecanismos que permitan facilitar la seguridad en la red, algunos sgbdr suministran herramientas que permiten garantizar la seguridad en un entorno de red. El sgbdr Oracle ofrece la herramienta Oracle Advanced Security.

Oracle Advanced Security [Ora07] protege la privacidad y confidencialidad de los datos de red al eliminar el espionaje de datos, la pérdida de datos, los ataques de intermediarios (person-in-the-middle) y de repetición (replay). Toda comunicación con una base de datos Oracle puede cifrarse con Oracle Advanced Security. Las bases de datos contienen información extremadamente sensible, y el acceso restringido mediante una estricta autenticación es una de las primeras líneas de defensa. Oracle Advanced Security brinda buenas soluciones de autenticación que aprovechan las estructuras de seguridad existentes de una empresa, con inclusión de Kerberos[36], Criptografía de Clave Pública[37], y RADIUS[38].

3.4.1.6 Monitorización

En el punto 10.10[39] del estándar se expone la necesidad de producir y mantener registros de auditoría de las actividades, excepciones y eventos de seguridad de la información durante un período acordado para ayudar en investigaciones futuras y permitir la monitorización del control de acceso.

3.4.1.6.1 Registro de auditoría

Control

Se deberían producir y mantener registros de auditoría de las actividades, excepciones y eventos de seguridad de la información durante un período acordado, para ayudar en investigaciones futuras y permitir la monitorización del control de acceso.

Directrices de implementación

Los registros de auditoría deberían incluir, cuando sea relevante:

- Utilizar Id's.
- Fechas, horas y detalles de eventos claves; por ejemplo, ingreso y salida.

[36] Kerberos [MIT12] es un protocolo de autenticación de red. Está diseñado para proveer autenticación fuerte para aplicaciones que siguen el modelo cliente/servidor utilizando criptografía de clave secreta. Una implementación libre de este protocolo está disponible en el Massachusetts Institute of Technology. Kerberos está disponible en una amplia gama de productos comerciales.

[37] [Her99] Empleando este sistema, cada usuario dispone de un par de claves: clave privada y clave pública. La clave privada es mantenida por el usuario en secreto mientras que la clave pública será conocida por todos aquellos que quieran intercambiar información con este usuario. A través de estas claves se podrá cifrar y descifrar el contenido de los mensajes intercambiados.

[38] RADIUS (Remote Authentication Dial-In User Server) [RWR+00] es un protocolo de autenticación y autorización para aplicaciones de acceso a la red con movilidad IP. Este protocolo emplea el puerto 1812 para establecer sus conexiones.

[39] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en páginas 55-57.

- Identidad o ubicación de la identidad, si es posible.
- Registros de intentos de acceso fallidos y rechazados al sistema.
- Registros de intentos de acceso fallidos y rechazados a los datos y otros recursos.
- Cambios en la configuración del sistema.
- Uso de privilegios.
- Uso de las utilidades y aplicaciones del sistema.
- Archivos a los cuales se tuvo acceso y los tipos de acceso.
- Direcciones y protocolos de la red.
- Alarmas activadas por el sistema de control de acceso.
- Activación y desactivación de los sistemas de protección; como sistemas anti-virus y sistemas de detección de intrusiones, aplicados consistentemente a través de la infraestructura de procesamiento de la información.

Otra información

Los registros de auditoría son susceptibles de contener datos personales que deberían ser tratados de forma confidencial. Se deberían mantener las medidas de protección de privacidad apropiadas. Cuando sea posible, los administradores del sistema no deberían tener posibilidad de borrar o desactivar los registros de sus propias actividades.

3.4.1.6.2 Registros del administrador y operador

Control

Se deberían registrar las actividades del administrador del sistema y del operador del sistema.

Directrices de implementación

Los registros deberían incluir:

- La hora en la cual ocurre un evento (éxito o fallo).
- La información sobre el evento (por ejemplo, archivos manejados) o fallo (por ejemplo el error ocurrido y la acción correctiva aplicada).
- Qué cuenta y qué operador o administrador está implicado.

- Qué procesos están involucrados.

- Los registros de administrador y operador del sistema deberían ser revisados de manera regular.

Otra información

Se puede utilizar un sistema de detección de intrusiones gestionado fuera del control del sistema. Los administradores del sistema podrían llevar a cabo labores de monitorización del sistema y de las actividades de administración de la red, para verificar su funcionamiento correcto.

3.4.1.6.3 Registro de fallos

Control

Se deberían registrar y analizar los fallos, y se deberían tomar las acciones necesarias.

Directrices de implementación

Se deberían registrar los fallos notificados por los usuarios o por los programas del sistema relacionados con los problemas relativos al procesamiento de la información o los sistemas de comunicación. Deberían existir reglas claras para manejar los fallos reportados incluyendo:

- Revisión de los registros de fallos para asegurar que los fallos se hayan resuelto satisfactoriamente.

- Revisión de las medidas correctivas para asegurar que los controles no se hayan visto comprometidos, y que la acción tomada haya sido completamente autorizada.

Se debería asegurar que el registro de errores está activado, si está disponible esta función del sistema.

Otra información

Los registros de errores y fallos pueden tener un impacto en el rendimiento del sistema. Este registro debería ser facilitado por el personal competente, y se debería determinar el nivel de registro requerido para los sistemas individuales mediante una evaluación del riesgo, tomando en cuenta la degradación del rendimiento.

3.4.1.6.4 Implementación en un sgbdr

Algunos sgbdr disponen de herramientas que permiten la generación de registros de auditoría. Estos registros podrían facilitar información relativa a preguntas importantes como:

- Quién lo hizo.
- Qué hizo.
- Dónde lo hizo (sobre qué esquema u objeto).
- Cuándo lo hizo.
- Cómo lo hizo (que sentencia SQL utilizada).

Particularizando la utilización de este tipo de mecanismos sobre el sgbdr de Oracle aparecen las siguientes posibilidades:

- Auditoría Estándar [Ben09]: constituye el mecanismo más exhaustivo y completo que ofrece Oracle. Permite auditar actividades basándose en el tipo de actividad, objeto, privilegio o usuario. Cuando se utiliza este tipo de mecanismo se debe llevar a cabo en dos fases: la primera consistirá en habilitar este tipo de auditoría y la segunda será la definición de categorías para auditar (por ejemplo definir qué actividades deberían generar registros de auditoría).

- Auditoría de grano fino [Ben09]: este tipo de auditoría se añadió al sgbdr de Oracle para habilitar el establecimiento de requisitos de auditoría que permitieran explícitamente definir condiciones bajo las que debería crearse un registro de auditoría. Este tipo de condiciones podrían estar basadas en datos concretos dentro de ciertas columnas o en columnas concretas que han sido accedidas. La diferencia con respecto a la auditoría estándar radica en que esta se basa en comandos y objetos (o se audita la totalidad o no se audita). La auditoría de grano fino permite, sin embargo, especificar condiciones particularizadas que determinan si se debe generar o no un registro de auditoría.

- Oracle Audit Vault [Ben09]: constituye un producto separado del sgbdr. Esta aplicación es utilizada para gestionar políticas de auditoría y para permitir recoger los registros de auditoría de varias bases de datos. No es un tipo de herramienta que añada nuevas capacidades de auditoría sino que lo que permite es ubicar la totalidad de registros de auditoría utilizables por aplicaciones para la generación de informes y alertas asociadas.

La utilización de este tipo de mecanismos sobre el sgbdr Oracle podría tener un impacto importante sobre el rendimiento del sistema por lo que se debe evaluar con antelación la cantidad de elementos que se desean auditar, así como realizar pruebas previas a la implantación de cualquier cambio en el perfil de auditoría.

3.4.1.7 Gestión de acceso de los usuarios

En el punto 11.2[40] del estándar se expone la necesidad de asegurar el acceso del usuario autorizado y evitar el acceso no autorizado a los sistemas de información.

Se deberían establecer procedimientos formales para controlar la asignación de los derechos de acceso a los sistemas y servicios de información.

Los procedimientos deberían abarcar todas las etapas en el ciclo de vida del acceso del usuario, desde el registro inicial de usuarios nuevos hasta la baja de los usuarios que ya no requieren acceso a los sistemas y servicios de información. Cuando sea apropiado, se debería prestar especial atención a la necesidad de controlar la asignación de derechos de acceso privilegiados, que son aquellos que permiten a los usuarios superar los controles del sistema.

3.4.1.7.1 Registro del usuario

Control

Debería existir un procedimiento formal para el registro y la salida del usuario con el objetivo de otorgar y revocar el acceso a todos los sistemas y servicios de información.

Directrices de implementación

El procedimiento de control del acceso para el registro y la salida del usuario del sistema debería incluir:

- Utilizar Id's de usuarios únicos para permitir a los usuarios vincularse y ser responsables de sus acciones; únicamente se debería permitir el uso de Id's grupales cuando son necesarios por razones comerciales u operacionales, y deberían ser aprobados y documentados.

- Chequear que el usuario tenga la autorización concedida por el propietario del sistema para el uso del sistema o servicio de información; también podría ser adecuada una aprobación por separado de la gerencia para los derechos de acceso.

- Chequear que el nivel de acceso otorgado sea apropiado para el propósito comercial y que sea consistente con la política de seguridad de la organización; por ejemplo, no compromete la segregación de responsabilidades.

- Proporcionar a los usuarios un listado por escrito de sus derechos de acceso.

[40] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en página 60.

- Requerir a los usuarios que firmen el listado indicando que entienden las condiciones de acceso.

- Asegurar que a los proveedores del servicio no se les proporciona acceso hasta que se hayan completado los procedimientos de autorización.

- Mantener un registro formal de todas las personas registradas para usar el servicio.

- Eliminar o bloquear inmediatamente los derechos de acceso de los usuarios que han cambiado de puesto o trabajo o han dejado la organización.

- Chequeo periódico para eliminar o bloquear los Id's de usuario y cuentas redundantes.

- Asegurar que no se asignen Id's de usuario redundantes a otros usuarios.

Otra información

Se debería considerar establecer roles de acceso de usuarios basados en los requerimientos que agrupen un número de privilegios de acceso en perfiles de acceso de usuario típicos. Las solicitudes y revisiones relativas al acceso son más fáciles de manejar en el nivel de dichos roles en lugar de en el nivel de privilegios particulares.

Se debería considerar incluir en los contratos del personal y contratos de servicio cláusulas que especifiquen las sanciones si el personal o los agentes de servicio intentan accesos no autorizados.

3.4.1.7.2 Gestión de privilegios

Control

Se debería restringir y controlar la asignación y uso de privilegios.

Directrices de implementación

Los sistemas multi-usuario que requieren protección contra el acceso no autorizado deberían controlar la asignación de privilegios a través de un proceso de autorización formal. Se deberían considerar los siguientes pasos:

- Los privilegios de acceso asociados con cada producto que compone el sistema; por ejemplo, sistema operativo, sistema de gestor de base de datos y cada aplicación, y se deberían identificar los usuarios a quienes se les necesita asignar estos privilegios.

- Los privilegios se deberían asignar a los usuarios sobre la base de "sólo lo que necesitan saber" y sobre una base de evento-por-evento en línea con la política de control del acceso; es decir, los requerimientos mínimos para su rol funcional, sólo cuando se necesitan.

- Se debería mantener un proceso de autorización y un registro de todos los privilegios asignados. No se debería otorgar privilegios hasta que se complete el proceso de autorización.

- Se debería promover el desarrollo y uso de rutinas del sistema para evitar la necesidad de otorgar privilegios a los usuarios.

- Se debería promover el desarrollo y uso de los programas que evitan la necesidad de ejecutarse con privilegios.

- Los privilegios se deberían asignar a un Id de usuario diferente de aquellos utilizados para el uso normal del negocio.

Otra información

El uso inapropiado de los privilegios de administración del sistema (cualquier dispositivo o medio de un sistema de información que permite al usuario superar los controles del sistema o aplicación) puede ser un factor que contribuye mucho a los fallos o problemas de seguridad en el sistema.

3.4.1.7.3 Gestión de las claves secretas de los usuarios

Control

La asignación de claves secretas se debería controlar a través de un proceso de gestión formal.

Directrices de implementación

El proceso debería contemplar los siguientes requerimientos:

- Se debería requerir que los usuarios firmen una declaración para mantener confidenciales las claves secretas y mantener las claves secretas de grupo sólo dentro de los miembros el grupo; esta declaración firmada se puede incluir en los términos y condiciones del empleo.

- Cuando se requiere que los usuarios mantengan sus propias claves secretas, inicialmente se les debería proporcionar una clave secreta temporal segura, la cual estarán obligados a cambiar inmediatamente.

- Establecer procedimientos para verificar la identidad de un usuario antes de proporcionar una clave secreta nueva, substituta o temporal.

- Las claves secretas temporales deberían ser proporcionadas a los usuarios de una manera segura, se debería evitar el uso de mensajes de correo electrónico de terceros o no protegidos (texto sin cifrar).

- Las claves secretas temporales deberían ser únicas para la persona y no deberían ser fáciles de adivinar.

- Los usuarios deberían confirmar la recepción de las claves secretas.

- Las claves secretas nunca deberían ser almacenadas en los sistemas de cómputo de una forma desprotegida.

- Las claves secretas preestablecidas por el vendedor deberían ser cambiadas después de la instalación de sistemas o software.

Otra información

Las claves secretas son un medio común para verificar la identidad del usuario antes de otorgar acceso a un sistema o servicio de información en concordancia con la autorización del usuario. Están disponibles, y se debería considerar la idoneidad de otras tecnologías para la identificación y autenticación del usuario; tales como biométricas, por ejemplo verificación de huellas digitales, verificación de firmas; y el uso de dispositivos de hardware como tarjetas inteligentes.

3.4.1.7.4 Implementación en un sgbdr

Los sgbdr proporcionan mecanismos para controlar el registro de usuarios y permitir administrar los privilegios asociados. Limitándonos al ámbito del sgbdr de Oracle se puede afirmar que la seguridad del sistema se gestiona a través del concepto de usuario de base de datos. Para conectarse a una base de datos se necesita una cuenta válida de usuario. Los privilegios de los que se dispone dentro de la base de datos están basados en los privilegios que se asignan al usuario cuando se conecta a Oracle. La seguridad de esta cuenta de usuario es además uno de los bloques de construcción para garantizar un entorno de Oracle seguro.

Las contraseñas son el método más común con el que Oracle permite autentificarse cuando se entra en una base de datos. Asegurarse de que los usuarios utilizan contraseñas “fuertes” es una de los elementos más importantes que se deben controlar para proteger una base de datos.

A continuación se describen varios métodos de autenticación que utiliza Oracle:

- Autenticación mediante password [Orasite10]: cuando un usuario conecta con una base de datos se verifica que este usuario y la contraseña introducida almacenada en la base de datos, sea correcta. Las contraseñas se guardan cifradas en la base de datos (en el diccionario de datos).

- Autenticación externa [Orasite10]: en este caso es en el sistema operativo en el que se delega la autenticación. Cuando un usuario conecta con la base de datos se verifica que el nombre de usuario es el mismo que el nombre de usuario del sistema operativo para permitir la validación. En este caso no se almacenan las cuentas en la base de datos de ninguna forma. Estas cuentas están siempre precedidas del prefijo *OS$* .A partir de la versión 10g es posible configurar el parámetro *OS_AUTHENT_PREFIX*. A través de la opción *IDENTIFIED EXTERNALLY* se especifica al sgbdr que la cuenta es externa y tiene que ser validada por el sistema operativo.

- Autenticación global [Orasite10]: en este caso, cuando un usuario se conecta con la base de datos se verifica globalmente utilizando la opción avanzada de seguridad (*ADVANCED SECURITY OPTION*) para la autenticación sirviéndose de herramientas como Kerberos, RADIUS....Para las cuentas globales no se almacena tampoco nada en la base de datos. Mediante la opción *IDENTIFIED GLOBALLY*[41] se especifica al sgbdr que la cuenta se valida globalmente, mediante la utilización opción de seguridad avanzada.

3.4.1.8 Controles criptográficos

En el punto 12.3[42] del estándar se contempla la necesidad de proteger la confidencialidad, autenticidad o integridad a través de medios criptográficos.

Se debería desarrollar una política sobre el uso de controles criptográficos. La gestión de las claves debería ser adecuada para sostener el uso de técnicas criptográficas.

3.4.1.8.1 Política sobre el uso de controles criptográficos

Control

Se debería desarrollar e implementar una política sobre el uso de controles criptográficos para proteger la información.

Directrices de implementación

Cuando se desarrolla una política criptográfica se debería considerar lo siguiente:

- El enfoque de la gerencia sobre el uso de los controles criptográficos a través de la organización, incluyendo los principios generales bajo los cuales se debería proteger la información comercial.

[41] Tanto la opción *IDENTIFIED EXTERNALLY* como la opción *IDENTIFIED GLOBALLY* son utilizables con la sentencia *CREATE USER*.
[42] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en página 73.

- En base a la evaluación del riesgo, se debería identificar el nivel de protección requerido tomando en cuenta el tipo, fuerza y calidad del algoritmo criptográfico a seleccionar.

- El uso de codificación para la protección de la información confidencial transportada por los medios y dispositivos móviles o removibles o a través de las redes de comunicación.

- El enfoque de la gestión de claves, incluyendo los métodos para lidiar con la protección de las claves criptográficas y la recuperación de la información codificada en el caso de claves pérdidas, comprometidas o dañadas.

- Roles y responsabilidades; por ejemplo, quién es responsable de:

    - la implementación de la política.
    - la gestión de claves, incluyendo la generación de claves.

- Los estándares a adoptarse para la implementación efectiva en toda la organización (qué solución se utiliza sobre qué procesos comerciales).

- El impacto de utilizar información codificada sobre los controles que se basan en la inspección del contenido (por ejemplo, detección de virus).

Cuando se implementa la política criptográfica de la organización, se deberían considerar las regulaciones y las restricciones nacionales que se podrían aplicar al uso de técnicas criptográficas en diferentes partes del mundo y los problemas del flujo entre fronteras de la información codificada.

Se pueden utilizar controles criptográficos para lograr diferentes objetivos de seguridad:

- Confidencialidad: utilizando la codificación de la información para proteger la información confidencial o crítica, ya sea almacenada o transmitida.

- Integridad/autenticidad: utilizando firmas digitales o códigos de autenticación del mensaje para proteger la autenticidad e integridad de la información confidencial o crítica almacenada o transmitida.

- No-repudiación: utilizando técnicas criptográficas para obtener prueba de la ocurrencia o no-ocurrencia de un evento o acción.

Otra información

La decisión de si es apropiada una solución criptográfica debería ser vista como parte de un proceso más amplio de evaluación del riesgo y selección de controles. El resultado de esta evaluación se puede utilizar para determinar si es apropiado un control criptográfico, qué tipo de control se debería aplicar, sobre qué propósitos y sobre qué procesos comerciales.

Es necesaria una política sobre el uso de controles criptográficos para maximizar los beneficios y minimizar los riesgos de utilizar técnicas criptográficas, y evitar el uso inapropiado o incorrecto. Cuando se utilizan firmas digitales, se debería considerar cualquier legislación relevante, en particular la legislación que describe las condiciones bajo las cuales una firma digital es aceptada legalmente.

Se debería buscar asesoramiento especializado para identificar el nivel de protección apropiado y definir las especificaciones adecuadas que proporcionarán la protección y el soporte requeridos para la implementación de un sistema de gestión de claves a utilizar.

ISO/IEC JTC1 SC27[43] ha desarrollado varios estándares relacionados con los controles criptográficos. Se puede encontrar mayor información en IEEE P1363[44] y las directrices sobre criptografía OCDE[45].

3.4.1.8.2 Gestión de claves

Control

Se debería establecer la gestión de claves para dar soporte al uso de técnicas criptográficas en la organización.

Directrices de implementación

Todas las claves criptográficas deberían estar protegidas contra una modificación, pérdida y destrucción. Además, las claves secretas y privadas necesitan protección contra la divulgación no autorizada. Se debería proteger físicamente el equipo utilizado para generar, almacenar y archivar las claves.

El sistema de gestión de claves se debería basar en un conjunto de estándares, procedimientos y métodos seguros acordados para:

- Generar claves para los diferentes sistemas criptográficos y las diversas aplicaciones.
- Generar y obtener certificados de claves públicas.
- Distribuir claves a los usuarios implicados, incluyendo el procedimiento de activación de estas claves, una vez recibidas.

[43] ISO/IEC JTC 1/SC 27 Tecnologías de la Información – Técnicas de Seguridad es un comité de estandarización dentro del comité técnico mixto ISO/IEC JTC 1 de la Organización Internacional de Estándares y la Comisión Electrotécnica Internacional [Hum04].

[44] El proyecto IEEE P1363 [IEEE08] desarrolla especificaciones estándar para la Criptografía de Clave Pública, con la finalidad de emitir una serie de documentos para el estándar IEEE.

[45] Directrices para una política criptográfica [OCDE97] .Recomendación del consejo en relación con las directrices para una política criptográfica. Organización para la Cooperación y el Desarrollo Económico.

- Almacenar claves, incluyendo el procedimiento con el que los usuarios autorizados obtienen acceso a las claves.

- Cambiar o actualizar las claves incluyendo las reglas sobre cuándo se deberían cambiar las claves y cómo se realiza este proceso.

- Gestionar claves comprometidas.

- Revocar las claves incluyendo tanto su retiro como su eliminación; por ejemplo, cuando las claves se han visto comprometidas o cuando el usuario deja la organización (en cuyos casos las claves deberían también ser archivadas).

- Recuperar las claves cuando han sido perdidas o corrompidas como parte de la continuidad y gestión del negocio; por ejemplo, para recuperar la información codificada.

- Archivar las claves; por ejemplo, para la información archivada o respaldada.

- Destruir las claves.

- Registrar y auditar las actividades relacionadas con la gestión de claves.

Para poder reducir la posibilidad de comprometer las claves, se deberían definir las fechas de activación y desactivación para que las claves sólo se puedan utilizar durante un período de tiempo limitado. El período de tiempo dependerá de las circunstancias bajo las cuales se está utilizando el control criptográfico, y el riesgo percibido.

Además del manejo seguro de las claves secretas y privadas, también se debería considerar la autenticidad de las claves públicas. Este proceso de autenticación se puede realizar utilizando certificados de claves públicas, los cuales normalmente son emitidos por una autoridad de certificación, la cual debería ser una organización reconocida con controles y procedimientos adecuados para proporcionar el grado de confianza requerido.

Los contenidos de los acuerdos o contratos de nivel de servicio con los proveedores externos de servicios de criptografía; por ejemplo, una autoridad de certificación; deberían abarcar los temas de responsabilidad, confiabilidad de los servicios y tiempos de respuesta para la provisión de los servicios.

Otra información

La gestión de las claves criptográficas es esencial para el uso efectivo de las técnicas criptográficas. La norma ISO/IEC 11770[46] proporciona información más detallada sobre la gestión de las claves. Los dos tipos de técnicas criptográficas son:

[46] ISO/IEC 11770-1:2010 [Cat10] define un modelo general de gestión de claves que es independiente de cualquier algoritmo criptográfico particular. Sin embargo, ciertos mecanismos de distribución de claves pueden depender de las propiedades de un algoritmo concreto, por ejemplo, propiedades de los algoritmos asimétricos.

- Técnicas de claves secretas, donde dos o más partes comparten la misma clave y esta clave es utilizada tanto para codificar como descodificar la información. Esta clave debería mantenerse en secreto ya que cualquiera que tenga acceso a la clave puede decodificar toda la información codificada con esa clave o puede introducir información no-autorizada utilizando la clave.

- Técnicas de claves públicas, donde cada usuario tiene un par de claves, una clave pública (que puede ser revelada a cualquiera) y una clave privada (que se tiene que mantener en secreto); se pueden utilizar las técnicas de claves públicas para la codificación y para producir firmas digitales (ver también ISO/IEC 9796[47] y ISO/IEC 14888[48]).

Existe la amenaza de la falsificación de la firma digital, reemplazándola con la clave pública de un usuario. El problema es tratado mediante el uso de un certificado de clave pública.

Las técnicas criptográficas también se pueden utilizar para proteger las claves criptográficas.

Tal vez se necesite considerar procedimientos para el manejo legal del acceso a las claves criptográficas; por ejemplo, tal vez se necesite que la información codificada esté disponible en una forma descodificada como evidencia en un caso ante un tribunal.

3.4.1.8.3 Implementación en un sgbdr

Algunos sgbdr habilitan la posibilidad de utilizar algoritmos de cifrado en distintas partes con la finalidad de dotar de seguridad al sistema. El sgbdr de Oracle utiliza diversos algoritmos de cifrado y resumen a distintos niveles. A continuación se enuncian distintos ejemplos [Ben09]:

- Durante el instante de autenticación de un usuario en Oracle 11g se utiliza un esquema denominado O5LOGON. Este proceso de autenticación del usuario en una base de datos utiliza el algoritmo hash de seguridad (SHA-1).

- En Oracle 11g las palabras clave son cifradas utilizando Advanced Encryption Standard (AES) antes de ser enviadas a través de la red (independientemente de si está o no habilitado un método de cifrado en la red).

[47] ISO / IEC 9796-2:2002 [Cat02] especifica tres esquemas de firma digital que permiten la recuperación de mensajes, de los cuales dos son deterministas (no aleatorio) y uno de ellos es aleatorio. La seguridad de los tres esquemas se basa en la dificultad de factorizar números grandes. Los tres esquemas pueden proporcionar tanto una recuperación total del mensaje como parcial.

[48] ISO / IEC 14888 [Cat08] especifica la firma digital con apéndice (cuando el proceso de verificación utiliza el mensaje como parte de la entrada, el mecanismo se denomina "mecanismo de firma con apéndice". Una función hash es la que se utilizada en el cálculo del apéndice). ISO / IEC 14888-1:2008 especifica los principios y los requisitos generales para la firma digital con apéndice. ISO / IEC 14888-2 aborda firmas digitales basados en factorización entera, e ISO / IEC 14888-3 aborda firmas digitales basadas en logaritmo discreto.

- Los algoritmos de cifrado de datos que Oracle soporta son los siguientes:

  - Algoritmos DES: algoritmo de cifrado de clave simétrica.
  - Triple DES: algoritmo más seguro que DES puesto que aplica un triple cifrado DES.
  - RSA: algoritmo de cifrado de clave pública o asimétrica.

3.4.1.9 Cumplimiento de los requerimientos legales

En el punto 15.1[49] del estándar se establecen como requisitos de obligado cumplimiento evitar las violaciones a cualquier ley; regulación estatutaria, reguladora o contractual; y cualquier requerimiento de seguridad.

El diseño, operación, uso y gestión de los sistemas de información pueden estar sujetos a requerimientos de seguridad estatutarios, reguladores y contractuales.

Se debería obtener el asesoramiento sobre los requerimientos legales específicos de los asesores legales de la organización o profesionales legales calificados adecuados. Los requerimientos legislativos varían de un país a otro y pueden variar para la información creada en un país que es transmitida a otro país (es decir, flujo de datos entre-fronteras).

3.4.1.9.1 Identificación de la legislación aplicable

Control

Se debería definir explícitamente, documentar y actualizar todos los requerimientos estatutarios, reguladores y contractuales relevantes, y el enfoque de la organización para satisfacer esos requerimientos, para cada sistema de información y la propia organización.

Directrices de implementación

Se deberían definir y documentar los controles y responsabilidades individuales específicos para satisfacer estos requerimientos.

3.4.1.9.2 Protección de los datos y privacidad de la información personal

Control

Se debería asegurar la protección y privacidad de los datos conforme lo requiera la legislación, regulaciones y, si fuesen aplicables, las cláusulas contractuales relevantes.

[49] Información extraída del estándar ISO/IEC 27002 [ISO/IEC05] en páginas 99-101.

Directrices de implementación

Se debería desarrollar e implementar una política de protección y privacidad de los datos. Esta política debería ser comunicada a todas las personas involucradas en el procesamiento de la información personal.

El cumplimiento de esta política y toda legislación y regulación de protección de datos relevante requiere una apropiada estructura y control de la gerencia. Con frecuencia, esto se logra asignando a una persona como responsable, por ejemplo un funcionario de protección de datos, quien debería proporcionar directrices a los gerentes, usuarios y proveedores de los servicios sobre sus responsabilidades individuales y los procedimientos específicos que se deberían seguir.

La responsabilidad asociada al manejo de la información personal y el refuerzo del conocimiento de los principios de protección datos deberían ser tratados en concordancia con la legislación y las regulaciones relevantes. Se deberían implementar las medidas técnicas y organizacionales adecuadas para protección la información personal.

Otra información

Un número de países han introducido legislación colocando controles sobre la recolección, procesamiento y transmisión de datos personales (generalmente la información relativa a personas vivas que pueden ser identificadas mediante esa información). Dependiendo de la legislación nacional respectiva, dichos controles pueden imponer impuestos a aquellos que recolectan, procesan y difunden información personal; y pueden restringir la capacidad para transferir los datos a otros países.

3.4.1.9.3 Relación con la LOPD

En el caso de la legislación española, con respecto al tratamiento de datos de carácter personal, se aplica la LOPD y en concreto asociado a este punto, para establecer una relación posterior con los sgbdr, se destaca el Artículo 9 que fija la obligación del responsable y del encargado del tratamiento de adoptar las medidas técnicas y organizativas necesarias para evitar la alteración, la pérdida, el tratamiento o el acceso no autorizado a los datos personales. Dichas medidas deberán tener en cuenta el estado de la tecnología, la naturaleza de los datos que deban protegerse y los riesgos a los que los mismos están sometidos.

3.4.1.9.4 Implementación en un sgbdr

En este punto se puede llevar a cabo una recopilación de la totalidad de medidas aplicables a un sgbdr puesto que la totalidad de las mismas están destinadas a evitar la alteración, la pérdida, el tratamiento o el acceso no autorizado a los datos y en particular son aplicables a los datos de carácter personal:

- Organización de la información para permitir restringir el acceso a personal no autorizado.

- Determinación y asignación de roles y responsabilidades bien definidas que impidan el acceso a información protegida a personal no autorizado.

- Eliminación de derechos de acceso cuando sea pertinente.

- Realización de copias de seguridad que garanticen la total disponibilidad de información.

- Aplicación de mecanismos que garanticen la protección de la información que circula a través de las redes de interconexión.

- Monitorización de la totalidad de eventos significativos relacionados con este tipo de datos que permita verificar tanto la corrección de las acciones como llevar a cabo un seguimiento de la actividad.

- Mecanismos de seguridad disponibles para garantizar el registro seguro de usuarios.

- Aplicación de algoritmos de criptografía que impidan el acceso a la información a personal no autorizado.

3.4.2 Metodología ISACA

La metodología de auditoría propuesta por la ISACA establece, que el primer paso a llevar a cabo, será fijar los objetivos de control que permitan minimizar los riesgos de control a los que está sometido un entorno de bases de datos. Por cada objetivo de control, y ateniéndose a la metodología propuesta por la ISACA, se deberán especificar las técnicas de control correspondientes a cada uno de estos objetivos. Una vez establecidas estas técnicas de control se propondrán una serie de pruebas de cumplimiento que permitirán verificar la consistencia de los mismos. A continuación, se realizará una exposición detallada de los objetivos de control relacionados con las bases de datos, que aparecen definidos en el COBIT [ITGI07].

3.4.2.1 PO2.Definir la arquitectura de la información

La función de los sistemas de información es permitir crear y mantener un modelo de información de negocio y definir los sistemas apropiados que permitan la utilización de esta información. Para llevar a cabo estos objetivos, se desarrollará un diccionario de datos corporativo que contendrá las reglas de sintaxis definidas por la organización, el esquema de clasificación de los datos y los niveles de seguridad establecidos. La definición de la arquitectura de la información permitirá incrementar la calidad del proceso de toma de decisiones, haciéndolo seguro y fiable, permitiendo que se suministre información segura.

Así mismo, este proceso posibilita la racionalización de los recursos de los sistemas de información para adaptarlos de forma apropiada a las estrategias del negocio. La aplicación de este proceso es, adicionalmente necesaria, para incrementar la solvencia sobre la integridad y la seguridad de los datos y permitir un mayor grado de efectividad y control en la compartición de información entre las aplicaciones y el resto de entidades susceptibles de intervenir en el proceso.

El proceso "Definir la arquitectura de la información" satisface el requerimiento del negocio de TI para:

- Agilizar la respuesta a los requerimientos.
- Proporcionar información confiable y consistente.
- Permitir la integración de forma transparente de las aplicaciones en los procesos de negocio.

Estos requisitos se satisfacen con el establecimiento de un modelo de datos empresarial, que incorpore el esquema de clasificación de los datos para asegurar la integridad y la consistencia de todos los datos.

Estos objetivos se logran:

- Asegurando la precisión de la arquitectura de la información y del modelo de datos.
- Determinando la propiedad de los datos.
- Clasificando la información usando un esquema de clasificación previamente establecido.

Esto se mide a través de:

- Porcentaje de elementos de datos redundados/duplicados.
- Porcentaje de aplicaciones que no cumplen con la metodología de arquitectura de la información establecida por la organización.
- Frecuencia de las actividades de validación de datos.

3.4.2.1.1 PO2.1 Modelo de arquitectura de información empresarial

Se debe establecer y mantener un modelo de información empresarial que facilite el desarrollo de aplicaciones y las actividades de soporte a la toma de decisiones, consistente con los planes de TI.

Este modelo debe facilitar la creación, el uso y el compartir de forma óptima la información empresarial de tal manera que se mantenga su integridad, sea flexible, funcional, rentable, oportuna, segura y tolerante a fallos.

3.4.2.1.2 PO2.2 Diccionario de Datos Empresarial y reglas de Sintaxis de Datos

Se debe mantener un diccionario de datos empresarial que incluya las reglas de sintaxis de datos de la organización. El diccionario facilita compartir elementos de datos entre las aplicaciones y los sistemas, fomenta un entendimiento común de datos entre los usuarios de TI y del negocio, y previene la creación de elementos de datos incompatibles.

3.4.2.1.3 PO2.3 Esquema de clasificación de datos

Se debe establecer un esquema de clasificación que pueda aplicarse a toda la organización, basado en la criticidad y sensibilidad de la información (pública, confidencial, secreta). Este esquema debe incluir detalles acerca de la propiedad de los datos, la definición de los niveles apropiados de seguridad y de controles de protección, y una breve descripción de los requerimientos de retención y destrucción de datos, además de un indicativo de su criticidad y sensibilidad. Este esquema se utilizará como base para aplicar controles como el control de acceso, el registro de actividad o el cifrado.

3.4.2.1.4 PO2.4 Administración de Integridad

Se deben definir e implementar procedimientos para garantizar la integridad y la consistencia de todos los datos almacenados en formato electrónico, tales como bases de datos, ficheros…

3.4.2.1.5 Cumplimiento del proceso "Definir la arquitectura de la información"

La administración del proceso "Definir la arquitectura de la información" que satisface el requerimiento de negocio de TI de agilizar la respuesta a los requerimientos, para brindar información confiable y consistente, y para integrar de forma transparente las aplicaciones hacia los procesos de negocio es:

- 0 No existente: cuando no existe conciencia de la importancia de la arquitectura de la información para la organización. El conocimiento, la experiencia y las responsabilidades necesarias para desarrollar esta arquitectura no existen en la organización.

- 1 Inicial/Ad Hoc[50]: cuando la gerencia reconoce la necesidad de una arquitectura de información. El desarrollo de algunos componentes de una arquitectura de información ocurre de manera Ad hoc. Las definiciones abarcan datos en lugar de información, y son impulsadas por ofertas de proveedores de software de aplicación. Existe una comunicación esporádica e inconsistente de la necesidad de una arquitectura de información.

[50] Expresión utilizada para referirse a algo que es adecuado sólo para un determinado fin.

- 2 Repetible pero intuitivo: cuando surge un proceso de arquitectura de información y existen procedimientos similares, aunque intuitivos e informales, que se siguen por distintos individuos dentro de la organización. Las personas obtienen sus habilidades al construir la arquitectura de información por medio de experiencia práctica y la aplicación repetida de técnicas. Los requerimientos tácticos impulsan el desarrollo de los componentes de la arquitectura de la información por parte de los individuos.

- 3 Definido: cuando la arquitectura de la información se entiende y se acepta, y la responsabilidad de su aplicación se asigna y se comunica de forma clara. Los procedimientos, herramientas y técnicas relacionados, aunque no son sofisticados, se han estandarizado y documentado y son parte de actividades informales de entrenamiento. Se han desarrollado políticas básicas de arquitectura de información, incluyendo algunos requerimientos estratégicos, aunque el cumplimiento de políticas, estándares y herramientas no se refuerza de manera consistente. Existe una función de administración de datos definida formalmente que establece estándares para toda la organización, y empieza a reportar sobre la aplicación y uso de la arquitectura de la información. Las herramientas automatizadas se empiezan a utilizar, aunque los procesos y reglas son definidos por los proveedores de software de bases de datos. Un plan formal de entrenamiento ha sido desarrollado, pero el entrenamiento real se basa en iniciativas individuales.

- 4 Administrado y medible: cuando se da soporte completo al desarrollo e implantación de la arquitectura de información por medio de métodos y técnicas formales. La responsabilidad sobre el desempeño del proceso de desarrollo de la arquitectura se refuerza y se mide el éxito de la arquitectura de información. Las herramientas automatizadas de soporte están ampliamente generalizadas, pero todavía no están integradas. Se han identificado métricas básicas y existe un sistema de medición. El proceso de definición de la arquitectura de información es proactivo y se orienta a resolver necesidades futuras del negocio. La organización de administración de datos está activamente involucrada en todos los esfuerzos de desarrollo de las aplicaciones, para garantizar la consistencia. Un repositorio automatizado está totalmente implementado. Se encuentran en implantación modelos de datos más complejos para aprovechar el contenido informativo de las bases de datos. Los sistemas de información ejecutiva y los sistemas de apoyo a la toma de decisiones[51] aprovechan la información existente.

- 5 Optimizado: cuando la arquitectura de información es reforzada de forma consistente a todos los niveles. El valor de la arquitectura de la información para el negocio se enfatiza de forma continua. El personal de TI cuenta con la experiencia y las habilidades necesarias para desarrollar y dar mantenimiento a una arquitectura de información robusta y sensible que refleje todos los

[51] En un sentido amplio, se definen los sistemas de apoyo a las decisiones (DSS por sus siglas en inglés *Decision Support System*) como un conjunto de programas y herramientas que permiten obtener oportunamente la información requerida durante el proceso de la toma de decisiones, en un ambiente de incertidumbre.

requerimientos del negocio. La información provista por la arquitectura se aplica de modo consistente y amplio. Se hace un uso amplio de las mejores prácticas de la industria en el desarrollo y mantenimiento de la arquitectura de la información incluyendo un proceso de mejora continua. La estrategia para el aprovechamiento de la información por medio de tecnología de almacenamiento de datos y minería de datos están bien definidas. La arquitectura de la información se encuentra en mejora continua y toma en cuenta información no tradicional sobre los procesos, organizaciones y sistemas.

3.4.2.2 DS11.Administración de datos

Una administración de datos efectiva requiere de la identificación de requerimientos de datos. El proceso de administración de información también incluye el establecimiento de procedimientos efectivos para administrar la librería de medios[52], el respaldo, la recuperación de datos y la eliminación apropiada de medios. Una efectiva administración de datos ayuda a garantizar la calidad, oportunidad y la disponibilidad de la información del negocio.

El proceso de "Administración de datos" satisface el requerimiento de negocio de TI para optimizar el uso de la información y garantizar la disponibilidad de la información cuando se requiera. Este proceso debe centrarse en mantener la integridad, exactitud y disponibilidad de la información cuando se requiera. Esto se logra con:

- El respaldo de los datos y la prueba de recuperación.
- Administración del almacenamiento de datos local y remoto.
- Desechado de manera segura de datos y de equipos.

Esto es medible a través de:

- La satisfacción del usuario con la disponibilidad de los datos.
- Porcentaje de restauraciones exitosas de datos.
- Número de incidentes en los que tuvo que recuperarse datos sensibles tras desechar los medios en los que estaban almacenados.

3.4.2.2.1 DS11.1 Requerimientos del negocio para "Administración de datos"

Se debe verificar que todos los datos que se espera procesar se reciben y procesan completamente, de forma precisa y a tiempo, y que todos los resultados se entregan de acuerdo a los requerimientos del negocio. Las necesidades de reinicio y reproceso están soportadas.

[52] Registro de los diferentes medios (discos, papel, memorias no volátiles…) utilizados para el almacenamiento de información.

3.4.2.2.2 DS11.2 Acuerdos de almacenamiento y conservación

Se deben definir e implementar procedimientos para la clasificación, almacenamiento y retención de los datos, de forma efectiva y eficiente para conseguir los objetivos de negocio, la política de seguridad de la organización y los requerimientos regulatorios.

3.4.2.2.3 DS11.3 Sistema de administración de librerías de medios

Definir e implementar procedimientos para mantener un inventario de medios almacenados y archivados para asegurar su usabilidad e integridad.

3.4.2.2.4 DS11.4 Eliminación

Definir e implementar procedimientos para asegurar que los requerimientos de negocio para la protección de datos sensibles y de software se consiguen cuando se aplican procedimientos de eliminación software/hardware o de transferencia de datos.

3.4.2.2.5 DS11.5 Respaldo y restauración

Definir e implementar procedimientos de respaldo y restauración de los sistemas, aplicaciones, datos y documentación en línea con los requerimientos de negocio y el plan de continuidad.

3.4.2.2.6 DS11.6 Requerimientos de seguridad para la Administración de Datos

Definir e implementar las políticas y procedimientos para identificar y aplicar los requerimientos aplicables a la recepción, procesamiento, almacenamiento y salida de datos para conseguir los objetivos de la organización, el cumplimiento de las políticas de seguridad de la organización y la normativa legal vigente.

3.4.2.2.7 Cumplimiento del proceso "Administración de datos"

La administración del proceso de "Administrar los datos" que satisface el requerimiento de negocio de TI de optimizar el uso de la información y garantizar la disponibilidad de la información cuando se requiera es:

- 0 No existente: cuando los datos no son reconocidos como parte de los recursos y los activos de la empresa. No está asignada la propiedad sobre los datos o sobre la responsabilidad de su administración. La calidad y la seguridad de los datos es deficiente o inexistente.

- 1 Inicial/Ad Hoc: cuando la organización reconoce la necesidad de una correcta administración de los datos. Hay un método adecuado para especificar requerimientos de seguridad en la administración de datos, pero no hay procedimientos implementados de comunicación formal. No se habilitan específicamente procedimientos sobre la administración de datos. La responsabilidad sobre la administración de los datos no es clara. Los procedimientos de respaldo y recuperación y los acuerdos sobre la eliminación de medios son adecuados.

- 2 Repetible pero intuitivo: cuando, a lo largo de toda la organización, existe conciencia sobre la necesidad de una adecuada administración de los datos. A un alto nivel empieza a observarse la propiedad o responsabilidad sobre los datos. Los requerimientos de seguridad para la administración de los datos son documentados por las personas adecuadas. Se llevan a cabo labores de monitorización dentro de TI sobre algunas actividades claves dentro de la función de administración de datos (respaldo, recuperación y desecho). Las responsabilidades para la administración de datos son asignadas de manera informal al personal correcto dentro de TI.

- 3 Definido: cuando se entiende y acepta la necesidad de la administración de datos, tanto dentro de TI como a lo largo de toda la organización. Se establece la responsabilidad sobre la administración de los datos. Se asigna la propiedad sobre los datos a la parte responsable que controla la integridad y la seguridad. Los procedimientos de administración de datos se formalizan dentro de TI y se utilizan algunas herramientas para respaldos/recuperación y desecho de equipos. Se lleva a cabo algún tipo de proceso de monitorización sobre la administración de datos. Se definen métricas básicas de desempeño. Comienza a aparecer el entrenamiento sobre administración de información.

- 4 Administrado y medible: cuando se entiende la necesidad de la administración de los datos y las acciones requeridas son aceptadas a lo largo de toda la organización. La responsabilidad de la propiedad y la administración de los datos están definidas, asignadas y comunicadas de forma clara en la organización. Los procedimientos se formalizan y son ampliamente conocidos, el conocimiento se distribuye y se comparte. Comienza a aparecer el uso de herramientas. Se acuerda con los clientes los indicadores de desempeño y meta, y se monitorizan por medio de un proceso bien definido. Se lleva a cabo entrenamiento formal para el personal destinado a la administración de los datos.

- 5 Optimizado: cuando se entiende y acepta dentro de la organización la necesidad de realizar todas las actividades requeridas para la administración de datos. Las necesidades y los requerimientos futuros son explorados de manera proactiva. Las responsabilidades sobre la propiedad de los datos y la administración de los mismos están establecidas de forma clara, se conocen ampliamente a lo largo de la organización y se actualizan periódicamente. Los procedimientos se formalizan y se difunden ampliamente, la compartición del conocimiento es una práctica estándar. Se utilizan herramientas sofisticadas con un máximo de automatización de la administración de los datos. Se acuerda con los clientes los indicadores de desempeño y metas, se asocian con los objetivos del negocio y se monitorizan de manera regular utilizando un proceso bien definido. Se exploran constantemente oportunidades de mejora. El entrenamiento y la formación del personal de administración de datos se institucionaliza.

3.5 Objetivos de control en el ciclo de vida de una base de datos

3.5.1 Introducción

A lo largo de los siguientes puntos se expondrán algunos objetivos y técnicas de control a tener en cuenta a lo largo del ciclo de vida de una base de datos (*Figura 8*) que abarca desde el estudio previo hasta su explotación.

Figura 8. Fases en el ciclo de vida de una base de datos establecidas en [PPP+08].

3.5.2 Estudio previo y plan de trabajo

En esta fase, deberá realizarse un estudio tecnológico de viabilidad en el cual se contemplen distintas alternativas para alcanzar los objetivos del proyecto. Este estudio debería acompañarse de un análisis coste-beneficio para cada una de las opciones consideradas. Entre las posibles alternativas contempladas debe considerarse la posibilidad de no abordar la solución implantando un sistema gestor de base de datos (no siempre está justificada la implantación de un sgbd).

Así mismo, debe considerarse la posibilidad de adquirir una solución previamente desarrollada que cubra los requisitos inicialmente establecidos (en la práctica, podría ocurrir que se ha desarrollado una aplicación cuya funcionalidad es equivalente a otra que ya existía previamente en el mercado y cuya adquisición hubiera supuesto asumir un riesgo inferior, asegurándose incluso una mayor calidad).

Si se lleva a cabo un proceso de auditoría durante el transcurso de esta fase, el auditor debe comprobar que la alta dirección revisa los estudios de viabilidad realizados y que es quien toma la decisión de seguir adelante o no con el proyecto.

Es importante destacar que desde el inicio de esta fase se debería aplicar una gestión de riesgos (valoración, identificación, medida, plan de acción y aceptación).

En el caso de que se decida llevar a cabo el proyecto, es fundamental que se establezca un plan director, sobre el que el auditor pueda comprobar que dicho plan se emplea para el seguimiento y para la gestión del proyecto, y que cumple con los procedimientos generales de gestión de proyectos que tenga aprobados la organización.

Un aspecto muy importante a considerar en esta fase es la aprobación de la estructura orgánica, no sólo de proyecto en particular, sino también de la unidad que tendrá la responsabilidad de la gestión y el control de la base de datos. En [ITGI07] se contemplan diferentes roles relacionados con la "gestión de datos". A la hora de detallar las responsabilidades de cada uno de estos roles, hay que tener en cuenta la explícita separación de funciones fundamentalmente entre:

- El personal de desarrollo de sistemas y el de explotación.
- Entre explotación y control de datos.
- Administración de bases de datos y desarrollo.

Adicionalmente, debería existir también una separación de funciones entre el administrador de la seguridad y el administrador de la base de datos. Esto no quiere decir que estas tareas tengan que ser desempeñadas por personas distintas pero sí que es un aspecto importante de control a considerar, por lo que en el caso de que no pueda lograrse la separación de funciones, deberán establecerse controles compensatorios o alternativos como, por ejemplo, una mayor atención de la dirección y la comprobación por parte de algún usuario del contenido y de las salidas más importantes producidas por la base de datos.

3.5.3 Concepción de la base de datos y selección del equipo

En esta fase se empieza a diseñar la base de datos, por lo que se deben utilizar los modelos y las técnicas definidos en la metodología de desarrollo de sistemas adoptada por la organización. La metodología de diseño debería también emplearse para especificar los documentos fuentes, los mecanismos de control, las características de seguridad y los elementos asociados con la auditoría que deben incluirse en el sistema a desarrollar. El auditor, en esta fase, debe analizar la metodología de diseño empleada con el fin de determinar si es adecuada. Tras esta comprobación, se debe verificar su correcta aplicación. Como mínimo la metodología de diseño de base de datos empleada debería contemplar dos fases de diseño: lógico y físico, aunque, en general la mayoría de las metodologías empleadas en la actualidad contemplan tres fases: conceptual, lógico y físico.

3.5.4 Diseño y carga

En esta fase se llevarán a cabo los diseños conceptual, lógico y físico de la base de datos, por lo que en este punto, el auditor tendrá que examinar si estos diseños se han realizado correctamente, determinando si la definición de los datos contempla además de su estructura, las asociaciones y las restricciones oportunas, así como las especificaciones de almacenamiento de datos y las cuestiones relativas a la seguridad. El auditor deberá tomar una muestra significativa de ciertos elementos (tablas, vistas e índices) y comprobar que su definición es completa, que ha sido aprobada por el usuario y que el administrador de la base de datos ha participado en algún instante, en este proceso.

Una vez diseñada la base de datos, se procederá a su carga, ya sea migrando los datos desde una fuente previa o introduciéndolos manualmente. Las migraciones o conversiones de sistemas, como el paso de un sistema de ficheros a uno de bases de datos, o de un determinado tipo de sgbd a otro (como ejemplo, de jerárquico a relacional) entrañan un riesgo muy importante por lo que deberán estar claramente planificadas para evitar pérdidas de información y la carga en el nuevo sistema de datos erróneos. También deben llevarse a cabo pruebas en paralelo, verificando que la decisión real de dar por terminada la prueba en paralelo se atiene a los criterios establecidos por la dirección y que se ha aplicado un control estricto de la corrección de errores detectados en esta fase.

Con respecto a la entrada manual de datos, se deben establecer un conjunto de controles que permitan asegurar la integridad de los mismos. Un aspecto muy importante a considerar es el tratamiento de datos de entrada erróneos, para los que deben cuidarse con atención los procedimientos de reintroducción de forma que no se disminuyan los controles para que al menos sean equivalentes a los originalmente establecidos.

En cualquier caso, los datos deben validarse y corregirse tan rápidamente como se detecte el problema.

Habitualmente, no toda la semántica de los datos puede almacenarse en el esquema de la base de datos, por lo que hay parte de la semántica que residirá en las aplicaciones. Será necesario comprobar que los programas implementan de forma adecuada esta integridad.

3.5.5 Explotación y mantenimiento

Una vez que se han realizado las pruebas de aceptación, con la participación de los usuarios, el sistema se pondrá (tras las correspondientes autorizaciones y siguiendo los procedimientos establecidos para ello) en explotación.

En este momento, deben verificarse que se establecen los procedimientos de explotación y mantenimiento que aseguren que los datos se tratan de forma congruente y exacta, y que el contenido de los sistemas solo se modifica mediante la aplicación de los procedimientos establecidos.

Sería conveniente también que el auditor pudiera llevar a cabo una auditoría sobre el rendimiento de la base de datos, comprobando si se ha realizado un proceso de ajuste (tuning[53]) y optimización adecuados, que no solo podría llegar a afectar al rediseño lógico y físico de la base de datos, sino que también podría comprender alterar ciertos parámetros del sistema operativo, e incluso la forma en la que se han diseñado las transacciones de la base de datos.

3.5.6 Revisión post-implantación

Esta fase implica la elaboración de un plan que permita efectuar una revisión post-implantación de todo el sistema, con el fin de evaluar si:

- Se han conseguido los resultados esperados.
- Se satisfacen las necesidades de los usuarios.
- Los costes y beneficios coinciden con los previstos.

[53] El término tuning se aplica a un proceso de adecuación de la configuración de los procesos asociados a un sgbd de tal forma que se consiga una mejora en el funcionamiento del sistema. Este proceso implica configurar estos procesos para que solo consuman los recursos necesarios sin que se vea afectada la velocidad de respuesta al realizar operaciones en la base de datos. En ocasiones, este proceso implica configurar la cantidad de bloqueos que se permiten a la vez, la cantidad de usuarios permitidos, etc.
Este proceso de optimización podría implicar la posibilidad de estructurar la base de datos en diferentes discos, es decir, que los diferentes volúmenes de información estén alojados en varios discos físicos distintos.

3.5.7 Otros procesos auxiliares

A lo largo de todo el ciclo de vida de la base de datos se deberá controlar la formación que precisan tanto los usuarios informáticos (administrador, analistas, programadores, etc.) como los no informáticos; ya que la formación es una de las claves para minimizar el riesgo en la implantación de una base de datos.

Esta formación no se puede basar simplemente en cursos sobre el producto que se está instalando, sino que suele ser precisa una formación de base, que resulta imprescindible cuando se pasa de trabajar de un entorno de ficheros orientado al proceso a un entorno de bases de datos, por lo que supone un "cambio filosófico", lo mismo puede decirse si se cambia de tipo de sgbd (por ejemplo, de relacional a orientado a objetos o semiestructurado).

Hay que tener en cuenta que usuarios poco formados constituyen uno de los riesgos más importantes de un sistema. Esta formación no debería limitarse al área de las bases de datos, sino que tendría que ser complementada con formación relativa a los conceptos de control y seguridad.

En este punto, el auditor tendrá que revisar la documentación que se produce a lo largo de todo el proceso, para verificar si es suficiente y si se ajusta a los estándares establecidos por la metodología adoptada en la empresa.

A este respecto, resulta muy importante que se haya llevado a cabo un aseguramiento de la calidad; lo ideal sería que en la propia empresa existiera un grupo de calidad que se encargara, entre otros aspectos, de asegurar la calidad de los diseños e implementaciones de bases de datos.

Capítulo 4

El Sistema Gestor de Bases de Datos Oracle 11g

4.1 Introducción

En este capítulo se llevará a cabo una descripción detallada del sistema gestor de bases de datos Oracle en su versión 11g centrándonos en su arquitectura, con el objetivo de definir, en capítulos posteriores, un procedimiento de auditoría que permita realizar un análisis lo más exhaustivo posible de todo el sistema, con la finalidad de detectar posibles debilidades y puntos de mejora.

4.2 Introducción al sgbdr Oracle 11g

El sistema gestor de bases de datos Oracle 11g constituye una implementación de un sistema gestor de bases de datos relacional (sgbdr) disponible para una gran variedad de plataformas (Unix, Linux, Windows). EL sgbdr Oracle 11g se comercializa de tres maneras diferentes [TB10]:

- Edición empresarial (enterprise edition).
- Edición estándar (standard edition) y edición estándar one (standard edition one).
- Edición personal (personal edition), únicamente disponible para plataformas Windows.

La edición empresarial está destinada a aplicaciones críticas dentro del entorno de una organización y contiene ciertas funcionalidades que permiten mejorar la disponibilidad y las capacidades de carga de las grandes bases de datos, además de facilitar su administración y optimización. Algunas de estas funcionalidades se enuncian a continuación:

- Oracle Real Application Clusters[54] (RAC): permite la utilización de Oracle en entornos clúster (alta disponibilidad, reparto de carga…).

- Oracle Partitioning: permite subdividir el almacenamiento de grandes objetos (tablas e índices) en varios elementos denominados particiones.

- Advanced Security Option: ofrece funcionalidades avanzadas con respecto a la gestión de la seguridad (criptografía, autenticación, etc.).

- Oracle Tuning Pack: módulo de administración que permite facilitar la optimización del rendimiento de la base de datos.

- Oracle OLAP y Oracle Data Mining: funcionalidades destinadas a la implantación de sistemas de apoyo a la toma de decisiones (DSS).

- Total Recall: solución que permite el almacenamiento a largo plazo de datos históricos.

- Real Application Testing: permite realizar capturas relativas a la actividad real de una base de datos y volcar esta actividad sobre otro sistema para realizar un análisis posterior.

- Advanced Compression: permite la compresión de todos los tipos de datos (estructurados o no estructurados).

La edición estándar comprende todas las funcionalidades principales que permiten poner en marcha aplicaciones basadas en modelos cliente/servidor dentro del ámbito de Internet o de una Intranet[55] y está concebida para ser utilizada por un grupo de trabajo o por un departamento dentro de una empresa. Esta edición no permite el funcionamiento de algunas opciones avanzadas de Oracle 11g y está limitada a servidores o clústeres con una capacidad máxima de cuatro procesadores. Adicionalmente, Oracle comercializa una edición estándar One, que en esencia, es idéntica a la edición estándar pero cuyo ámbito de aplicación está limitado a servidores con únicamente dos procesadores.

La edición personal es una versión mono licencia del producto, específicamente destinada a los desarrolladores. Ofrece el mismo nivel de funcionalidad que la edición empresarial.

[54] Oracle Real Application Clusters es una opción de la Edición Empresarial pero que está incluida, con pequeñas restricciones, en la Edición Estándar (no en la Edición Estándar One).
[55] Red propia de una organización, diseñada y desarrollada utilizando los protocolos propios de Internet, en particular la familia de protocolos TCP/IP. Puede tratarse de una red aislada, es decir no conectada a Internet.

4.3 Principales novedades en la versión 11g

El sistema gestor de bases de datos Oracle en su versión 11g aporta un gran número de novedades y mejoras en varios aspectos. A continuación, se enuncian las principales novedades con respecto a versiones anteriores [Har10]:

- Instalación y actualización simplificadas.
- Gestión completamente automática de la memoria total utilizada por la instancia.
- Compresión de tablas, incluso, en un entorno transaccional.
- Gestión simplificada de parámetros de inicialización (SPFILE).
- Gestión automática de la anulación activada por defecto.
- Gestión simplificada del espacio temporal.
- Nueva caché utilizada para almacenar el resultado de las consultas.
- Configuración de la base de datos, por defecto, más segura.
- Nueva herramienta para facilitar el diagnóstico de incidencias (Automatic Diagnostic Repository).
- Asistente para la recuperación de datos (Data Recovery Advisor).
- Mejora en la gestión de ficheros de actualización almacenados en RMAN (Recovery Manager).
- Mejora de rendimiento de las copias de seguridad comprimidas en RMAN.
- Extensión de las técnicas de flashback para la anulación de una transacción validada (Flashback Transaction).
- Bases de datos intercambiables entre Linux y Windows.
- Mejora en la detección de bloques corruptos.
- Mejora en la creación y recuperación de una copia de seguridad de larga duración.
- Copia de seguridad y recuperación paralelas de grandes ficheros de datos.
- Índices invisibles.

- Tablas de sólo lectura.
- Compresión de la totalidad de un fichero de exportación creado por Data Pump.
- Cifrado del fichero de exportación creado por Data Pump.
- Columnas virtuales.
- Operadores PIVOT y UNPIVOT.
- Nuevo asistente para la resolución de problemas relativos a una sentencia SQL (SQL Repair Advisor).
- Mejora de la interfaz de usuario en Oracle Enterprise Manager Database Control.
- Utilización de LogMiner a través de la interfaz gráfica de Oracle Enterprise Manager Database Control.
- Cifrado de un Tablespace.

4.4 Bases de la arquitectura Oracle

4.4.1 Concepto de instancia y base de datos

Un servidor Oracle se compone de dos elementos distintos, la instancia y la base de datos.

La instancia está constituida por una estructura de memoria compartida y un conjunto de procesos. Estos elementos están íntimamente relacionados aunque deben ser claramente diferenciados. La estructura de memoria compartida se denomina System Global Area (SGA) mientras que los procesos se dividen en:

- Procesos en segundo plano (background process) encargados de la gestión de la información almacenada en la instancia.
- Procesos de servidor (server process) encargados de atender las solicitudes realizadas por los procesos cliente.

Cada uno de estos procesos dispone de un espacio de memoria privado denominado Program Global Area (PGA).

Existe la posibilidad de que se ejecuten varias instancias simultáneamente en un mismo servidor. En este caso, cada una de las instancias dispondrá de su propia SGA y sus propios procesos.

Una instancia sólo puede abrir una base de datos al mismo tiempo y, en la gran mayoría de los casos, una base de datos es abierta por una única instancia. La utilización de la opción Real Application Clusters (RAC), permite que una base de datos pueda ser abierta por diferentes instancias situadas en nodos diferentes de un clúster de servidores.[56]

La instancia utiliza un fichero de parámetros durante su inicio que incluye su configuración y datos para manejar la base de datos.

Además de los procesos asociados a la instancia, existen procesos de usuario relativos a las aplicaciones que utilizan con la finalidad de manejar la base de datos. En una arquitectura cliente/servidor estos procesos estarán localizados en el puesto de usuario y se comunicaran con el servidor Oracle utilizando la capa Oracle Net[57].

La base de datos está constituida por un conjunto de ficheros físicos que se detallan a continuación:

- Uno o varios ficheros de datos que almacenan los datos propiamente dichos.
- Por, como mínimo, un fichero de control que contiene la información de control sobre la base de datos.
- Por, como mínimo, dos grupos de ficheros de actualización que registran todas las modificaciones realizadas en la base de datos. Estos ficheros de actualización pueden almacenarse.

Cada base de datos tiene un nombre definido en el momento de su creación. Este nombre se define por el parámetro de inicialización *DB_NAME* del fichero de configuración de parámetros.

[56] Esta opción podría ser interesante para sistemas en los que haya que ofrecer una alta disponibilidad.
[57] Oracle Net es una capa de software que permite a los productos Oracle comunicarse entre sí. Las funciones esenciales de Oracle Net incluyen el establecimiento de sesiones de comunicación entre dos máquinas (cliente-servidor o servidor-servidor) para permitir una transferencia de datos.

4.4.2 Arquitectura de la base de datos

En este apartado se llevará a cabo una recopilación de los diferentes tipos de ficheros que componen una base de datos describiendo de forma detallada su funcionalidad [Heu09].

4.4.2.1 Fichero de control

El fichero de control contiene la información de control de la base de datos:

- El nombre de la base de datos.
- La fecha/hora de creación de la base de datos.
- La ubicación de otros ficheros de la base de datos (ficheros de datos y ficheros de actualización).
- El número secuencial actual de los ficheros de actualización.
- Información de los puntos de comprobación (checkpoints), etc.

El sgbd actualiza el fichero de control automáticamente cada vez que se produce una modificación de la estructura de la base de datos (cambio de la ubicación de un fichero, por ejemplo).

En el instante que se ejecuta una instancia para abrir una base de datos, el fichero de control es el primero en abrirse. Permite a la instancia localizar y abrir el resto de ficheros de la base de datos. Si no se puede encontrar el fichero de control (o está dañado), la base de datos no puede abrirse, aunque el resto de ficheros de la base de datos esté presente.

Por razones de seguridad, es aconsejable replicar el fichero de control, es decir tener diferentes copias creadas en espejo. Técnicamente, es posible crear una base de datos con un único fichero de control pero es extremadamente aconsejable utilizar varias copias, aunque el servidor únicamente disponga de un disco.

En el instante de la creación de una base de datos, es posible especificar varios ficheros de control. Esta acción también se puede llevar a cabo después de su creación.

4.4.2.2 Ficheros de actualización

Los ficheros de actualización (redo log) registran todas las modificaciones realizadas en la base de datos. Se organizan en grupos que se sobrescriben de manera cíclica.

Los ficheros de actualización se utilizan para la recuperación de la instancia tras una parada anormal y para la recuperación si un fichero de datos se pierde o se daña; en este caso, la información contenida en los ficheros de actualización se aplica sobre una copia de seguridad de los ficheros de datos con el objetivo de restaurar la base de datos al estado e instante correctos.

Los ficheros de actualización se organizan en grupos (un mínimo de dos) compuestos de uno o varios miembros (mínimo uno). Estos grupos se generan en el instante de la creación de la base de datos. En el interior de un grupo, los miembros se escriben simultáneamente en espejo por la instancia sgbd (procesos LGWR) y contienen la misma información. Todos los miembros de un grupo tienen el mismo tamaño, definido en el momento de la creación de un grupo. Por tanto, un fichero de actualización contiene una cantidad máxima de información. El número de grupos se establece en el instante de la creación y no aumenta dinámicamente.

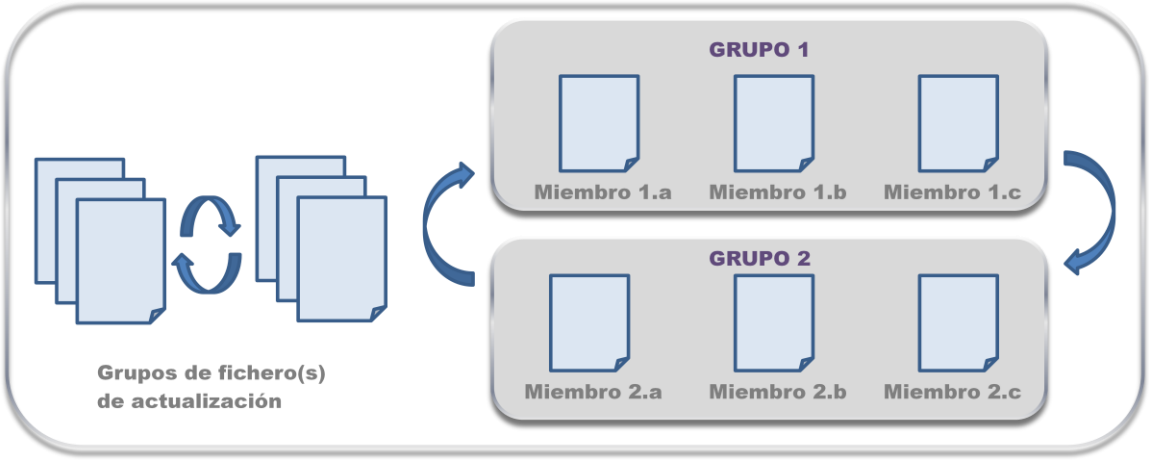


Figura 9. Representación de grupos de ficheros de actualización.

Cuando la instancia Oracle regresa al primer grupo, sobrescribe la información que se encuentra almacenada hasta ese instante. Por tanto, esta información no estará disponible en caso de necesidad. Para garantizar la posibilidad de realizar recuperaciones completas, es necesario activar un mecanismo de almacenamiento que permite salvaguardar los ficheros de actualización cuando están llenos, antes de que la instancia los reutilice[58].

Si un grupo contiene varios miembros y uno de ellos no está disponible, la base de datos puede continuar funcionando.

Los ficheros de actualización constituyen un elemento muy importante para la seguridad de la base de datos. Por lo tanto, se aconseja utilizar un mínimo de dos o tres miembros por grupo, si es posible, en discos diferentes.

4.4.2.3 Ficheros de datos

Los ficheros de datos contienen los datos propiamente dichos de la base de datos (tablas e índices fundamentalmente). Desde un punto de vista lógico, se agrupan en Tablespaces. Un Tablespace es una unidad lógica de almacenamiento compuesta por uno o varios ficheros físicos. La práctica totalidad de operaciones de administración relativas al almacenamiento, se efectúan trabajando sobre un Tablespace y no sobre los ficheros de datos subyacentes.

A partir de la versión 10g, Oracle introdujo la noción de Tablespace bigfile[59]. Un Tablespace bigfile es un Tablespace que sólo contiene un fichero de datos, pero que puede ser mucho más grande que un fichero de datos tradicional. Los Tablespaces bigfile permiten gestionar volúmenes de datos mucho más importantes, todo esto simplificando la gestión del almacenamiento (menos ficheros, transparencia del fichero de datos).

Una base de datos contiene un mínimo de dos ficheros de datos, pertenecientes a dos Tablespaces reservados para el sgbd de Oracle (el Tablespace *SYSTEM* y el Tablespace *SYSAUX* que apareció a partir de la versión 10g). Los Tablespaces *SYSTEM* y *SYSAUX* no deberían contener ningún dato funcional. Debido a esto, en la práctica, una base de datos contendrá una serie de ficheros de datos asociados a otros Tablespaces.

Con respecto a la organización del almacenamiento, los ficheros de datos se dividen en bloques de un tamaño preestablecido (4kb, 8kb…).

Al espacio ocupado por un determinado objeto en un Tablespace se le denomina a través del término genérico de segmento.

[58] Modo Archivelog. Este modo de funcionamiento implica que el sgbd activará un proceso encargado de llevar a cabo el almacenamiento (*ARC0* es la denominación del proceso encargado de realizar esta labor) del grupo de ficheros de actualización antes de que el proceso *LGWR* proceda a su reutilización. Con la base de datos configurada en modo *Archivelog* es posible la realización de backups con la base de datos arrancada, lo que permitirá llevar a cabo una restauración completa de la base de datos sin pérdida de datos.

[59] En parte de la bibliografía consultada se pueden encontrar referencias a los Tablespaces normales denominándolos smallfile para distinguirlos de los Tablespaces bigfile.

Hay cuatro tipos principales de segmentos:

- Segmentos de tablas: espacio ocupado por las tablas.
- Segmentos de índice: espacio ocupado por los índices.
- Segmentos de anulación: espacio temporal utilizado para almacenar la información que permite anular una transacción.
- Segmentos temporales: espacio temporal utilizado habitualmente en operaciones de ordenación.

Un segmento pertenece a un Tablespace y está constituido por extensiones[60] (extents). Una extensión es un conjunto de bloques contiguos en un fichero de datos.

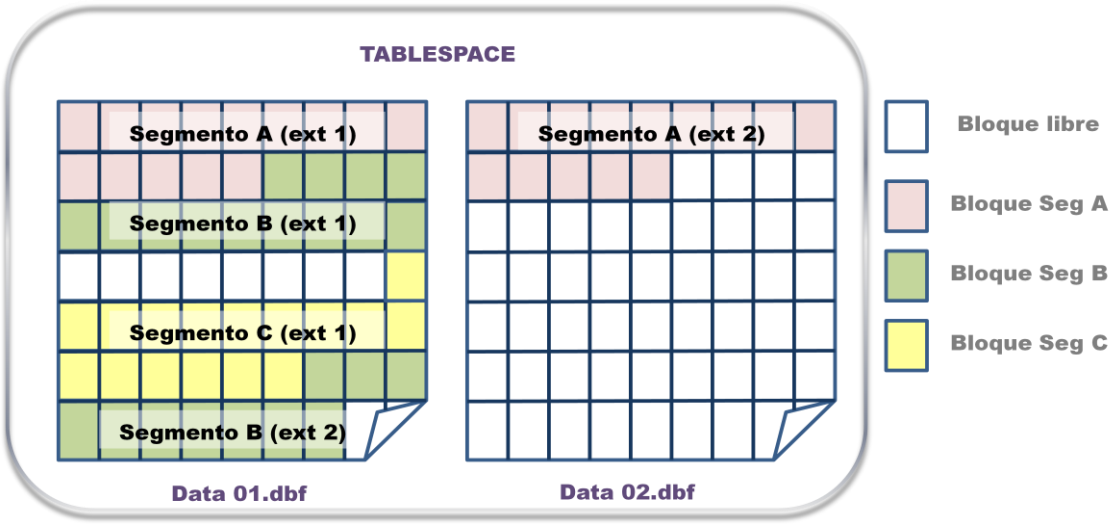


Figura 10. Representación de relaciones entre Tablespaces, segmentos y bloques.

Un bloque Oracle es la unidad más pequeña de entrada/salida utilizada por el sgbd Oracle. Todos los ficheros de datos se organizan en bloques Oracle y tienen un tamaño que es múltiplo del tamaño del bloque. El bloque Oracle es también la unidad de almacenamiento de la caché de datos (Database Buffer Cache) en la SGA. Cuando la instancia del sgbd de Oracle lee un fichero de datos, lee los bloques Oracle del fichero y los carga en los bloques Oracle de la caché de datos.

El segmento de anulación es una estructura utilizada por el sgbd de Oracle para el almacenamiento temporal de la versión anterior de los datos que están siendo modificados en una transacción. Si la transacción es válida (*COMMIT*), se libera el

[60] En Oracle Enterprise Manager, las extensiones se denominan "conjunto de bloques contiguos".

espacio ocupado. Si la transacción se anula (*ROLLBACK*), la versión anterior de los datos será restituida en lugar de la nueva.

A parte de los Tablespaces destinados a los datos propiamente dichos de aplicación (tablas, índices) necesitaremos crear dos Tablespaces "técnicos" utilizados internamente por el sgbd de Oracle: el Tablespace de anulación (para los segmentos de anulación) y el Tablespace temporal (para los segmentos temporales).

Cuando se crea un segmento (tabla, índice, etc.) se sitúa (explícitamente por aquel que lo crea o implícitamente por Oracle) en un Tablespace. Tras la creación, el sgbd suministra espacio a este segmento en uno de los ficheros de datos asociados al Tablespace.

En el instante de la creación de un segmento, se proporcionan una o varias extensiones. Cuando sus primeras extensiones están llenas (como consecuencia de una inserción de datos, por ejemplo), se proporciona una nueva extensión. Esta extensión se sitúa en el mismo Tablespace, aunque no forzosamente junto a la primera, ni siquiera tendría por qué estar en el mismo fichero de datos (si el Tablespace dispone de varios ficheros de datos). Cuando esta nueva extensión está llena, el proceso se repite.

En la sentencia SQL de creación del segmento, existen cláusulas que permiten indicar en qué Tablespace crear el segmento y definir el tamaño inicial del segmento.

Desde la versión 9i del sgbd de Oracle, es posible utilizar varios tamaños de bloque en la base de datos:

- Un tamaño de bloque "estándar" se define a través del parámetro de inicialización *DB_BLOCK_SIZE*.

- Se pueden utilizar hasta otros cinco tamaños de bloque: los valores permitidos son 2kb, 4kb, 8kb, 16kb o 32kb[61].

La posibilidad de utilizar varios tamaños de bloque es interesante a efectos de utilizar la funcionalidad de transporte de Tablespaces. Esta funcionalidad, aparecida a partir de la versión 8i, permite transportar un Tablespace de una base de datos origen y adjuntarlo a una nueva base de datos. Esta operación se efectúa gracias a la utilidad Data Pump, utilizando la opción *TRANSPORT TABLESPACES*. Uno de los prerrequisitos para la utilización de esta funcionalidad en Oracle 8i es que las bases de datos implicadas en el intercambio deben utilizar el mismo tamaño de bloque. Esta limitación desaparece a partir de la versión 9i ya que diferentes Tablespaces de una misma base de datos pueden utilizar tamaños de bloque diferentes: un Tablespace que utiliza tamaños de bloque de 4kb se puede transportar a una base de datos que utiliza bloques de 8kb.

[61] Determinadas plataformas son más restrictivas en la definición de los tamaños de bloque y no permiten fijar todos los tamaños de bloque inicialmente preestablecidos.

4.4.3 Arquitectura de la Instancia

En este punto [Har10] y [Heu09][62], se expondrá de manera detallada tanto la estructura de memoria que forma parte de la instancia como la funcionalidad desempeñada por los procesos en segundo plano y los procesos servidor que forman parte de la misma. En la siguiente figura, aparece un esquema básico asociado a los elementos que componen la instancia:

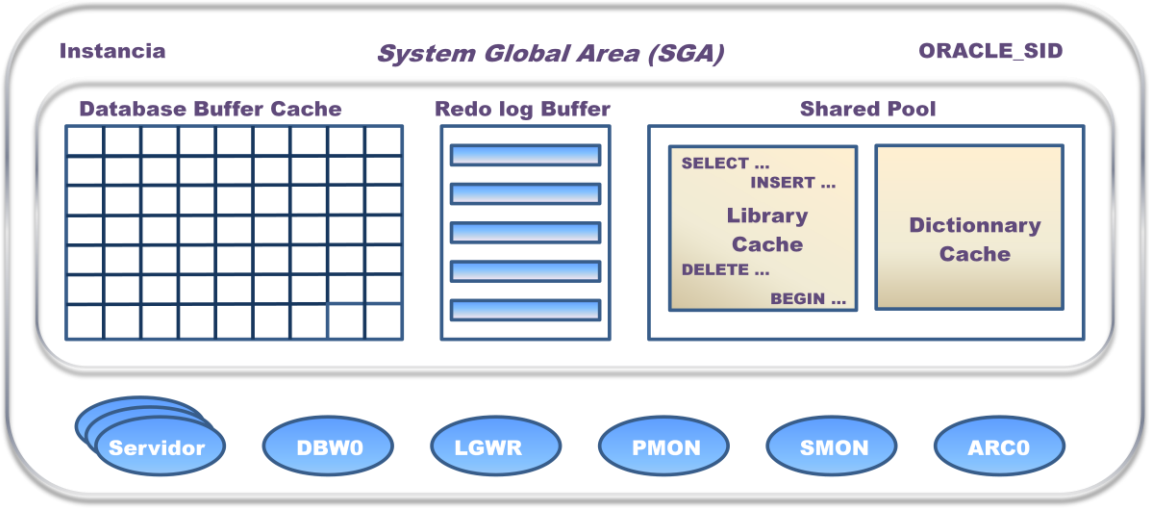


Figura 11.Elementos de la instancia.

4.4.3.1 SGA (System Global Area)

La SGA es una zona de memoria compartida por los diferentes procesos de la instancia. Esta zona de memoria se asigna en el inicio de la instancia y se libera en la parada de la misma. Se dimensiona por un conjunto de parámetros definidos en el fichero de parámetros.

Desde la versión 9i del sgbd de Oracle, la SGA se redimensiona durante la ejecución de la misma. Desde la versión 10g del sgbd de Oracle, ciertas estructuras de la SGA se pueden gestionar automáticamente.

El tamaño máximo de la SGA está limitado por el parámetro *SGA_MAX_SIZE*.

[62] Las ilustraciones que aparecen en este punto están basadas en esta bibliografía.

La SGA contiene las siguientes estructuras:

- Database Buffer Cache: caché de datos.
- Redo Log Buffer: memoria utilizada para el registro de las modificaciones realizadas en la base de datos.
- Shared Pool: zona de compartición de consultas y del diccionario de datos.
- Java Pool: memoria utilizada por la máquina virtual de Java integrada.
- Large Pool: zona de memoria opcional utilizada por diferentes procesos en configuraciones particularizadas.
- Streams Pool: zona de memoria utilizada por la funcionalidad Stream. Esta funcionalidad permite la circulación de la información entre procesos.
- Result Cache (aparecida desde la versión 11g del sgbd de Oracle): caché para almacenar el resultado de consultas SQL o funciones PL/SQL.

La SGA integra adicionalmente una estructura conocida como "SGA fija" que contiene la información relativa al estado de la base de datos y de la instancia con respecto a los bloqueos. Esta SGA fija no se redimensiona por el DBA. Su tamaño es reducido (algunos centenares de Kb).

4.4.3.1.1 Database Buffer Cache

Esta zona de memoria contiene los bloques de datos utilizados más recientemente:

- Bloques de tablas.
- Bloques de índices.
- Bloques de segmentos de anulación, conteniendo la versión anterior de los datos cuya modificación está en curso.

Estos bloques son leídos de la memoria por los procesos servidor y se escriben en los ficheros de datos por el o los procesos en segundo plano DBWn.

Toda modificación (INSERT, UPDATE, DELETE) de un bloque se efectúa en memoria y la escritura en disco es diferida.

La Database Buffer Cache es una caché de datos que juega el mismo papel que la Shared Pool, pero para los datos.

Los datos de la base de datos sólo son accesibles, en lectura o actualización, después de haber sido cargados en la Database Buffer Cache.

En la práctica, ya que la Database Buffer Cache tiene un tamaño limitado, el sgbd de Oracle utiliza el algoritmo LRU (Least Recently Used) para gestionar la caché: en el caso de que falte espacio, el sgbd de Oracle elimina de la caché los datos utilizados menos recientemente.

La Database Buffer Cache se dimensiona utilizando los siguientes parámetros:

- *DB_CACHE_SIZE*: tamaño de la caché para el tamaño de bloque por defecto (*pool DEFAULT*).

- *DB_nK_CACHE_SIZE*: tamaño de la caché para los bloques de n Kb (siendo n un valor entre 2, 4, 8, 16 o 32).

En el caso en el que la base de datos utilice Tablespaces que tengan tamaños de bloques diferentes al tamaño estándar, es necesario dimensionar los otros pools en la Database Buffer Cache, para los bloques en cuestión. Esta operación se efectúa gracias a los parámetros *DB_nK_CACHE_SIZE*.

En ciertas configuraciones más avanzadas, es posible también definir otros dos pools contenidos en la Database Buffer Cache:

- La Keep Pool está destinada al almacenamiento de datos que deben permanecer el mayor tiempo posible en memoria. Esta zona de memoria está dimensionada por el parámetro *DB_KEEP_CACHE_SIZE*.

- El Recycle Pool está destinada al almacenamiento de los datos que teóricamente no van a permanecer mucho tiempo en la memoria. Esta zona de memoria se dimensiona utilizando el parámetro *DB_RECYCLE_CACHE_SIZE*.

Antes de la versión del sgbd Oracle 9i, el tamaño de Database Buffer Cache estaba definido en número de bloques (de un tamaño definido por el parámetro *DB_BLOCK_SIZE*) utilizando el parámetro *DB_BLOCK_BUFFERS*. Este parámetro existe todavía por razones de compatibilidad pero actualmente no se utiliza.

4.4.3.1.2 Redo Log Buffer

Esta zona de memoria almacena información sobre las modificaciones en la base de datos, como paso previo a su escritura en uno de los ficheros de actualización.

Este buffer se utiliza de manera secuencial (las modificaciones de varias transacciones se mezclan) y cíclica (cuando está lleno, comienza de nuevo por el principio…después de haber sido escrito en el disco en un fichero de actualización).

Por cada modificación realizada, una entrada Redo (Redo Entry) se escribe en el buffer. Una entrada Redo está compuesta por un conjunto de vectores (Change Vector) que describen una modificación atómica de un bloque (tabla, índice o segmento de anulación). El nuevo valor, y también el antiguo, se registran en el fichero de actualización. De este modo, las tablas, los índices así como los segmentos de anulación están "protegidos" por los ficheros de actualización.

En el momento de una recuperación, los ficheros de actualización contienen la información necesaria para rehacer una transacción perdida o anular una transacción en curso en el momento del incidente.

Esta zona de memoria se dimensiona utilizando el parámetro *LOG_BUFFER*.

4.4.3.1.3 Shared Pool

La Shared Pool se compone principalmente de dos estructuras:

- Library Cache: Esta estructura contiene la información sobre las sentencias SQL y PL/SQL ejecutadas recientemente (texto de la sentencia, versión analizada, plan de ejecución[63]). Cuando una sentencia debe ser ejecutada, lo primero que debe hacer el sgbd de Oracle es analizarla (etapa de parser) para comprobar que es tanto sintáctica como semánticamente correcta para, como paso posterior, determinar el plan de ejecución de la sentencia. Debido a que esta etapa de análisis ocupa algo de tiempo y el resultado consume memoria, el sgbd de Oracle busca compartir las consultas entre diferentes usuarios con el objetivo de ganar tiempo y memoria si un usuario ejecuta una sentencia ya ejecutada con anterioridad. Cuando una consulta se ejecuta por primera vez, el sgbd de Oracle almacena el resultado del análisis en la Library Cache y después ejecuta la consulta. Cuando la misma consulta se ejecuta nuevamente más tarde, Oracle está en disposición de encontrarla en la Library Cache y ejecutarla directamente sin rehacer el análisis (o al menos, haciendo un análisis más ligero). En la práctica, puesto que la Library Cache tiene un tamaño limitado, el sgbd de Oracle utiliza un algoritmo LRU (Least Recently Used) para gestionar la caché: en el caso de que falte espacio, el sgbd de Oracle elimina de la caché la consulta utilizada menos recientemente.

- Dictionary Cache[64]: El diccionario de datos almacena toda la información relativa a la base de datos (lista de tablas y columnas, lista de usuarios y sus derechos, información de almacenamiento…). Durante la fase de análisis de una consulta, el sgbd utiliza el diccionario de datos para verificar que los objetos solicitados existen y que el usuario tiene el derecho de acceso para determinar donde se almacenan los objetos, etc. Para garantizar un buen nivel de rendimiento, el sgbd busca mantener todo o parte del diccionario de datos en la Dictionary Cache.

[63] El plan de ejecución define la secuencia de operaciones que el sgbdr realiza para ejecutar la sentencia SQL.
[64] Esta estructura de memoria aparece referenciada en cierta bibliografía como Row Cache.

La Shared Pool se dimensiona de manera global a través del parámetro de inicialización *SHARED_POOL_SIZE*. El sgbd asegura el reparto entre la Library Cache y la Dictionary Cache. El tamaño habitual de la Shared Pool está comprendido entre algunas decenas de Mb y algunas centenas de Mb.

4.4.3.1.4 Java Pool

La Java Pool se dimensiona por el parámetro *JAVA_POOL_SIZE* (24 Mb por defecto). A este parámetro se le puede asignar un valor 0 si la máquina virtual de Java integrada en Oracle no se utiliza. Esta zona de memoria es utilizada por algunas rutinas internas como por ejemplo las utilidades Import /Export.

4.4.3.1.5 Large Pool

Esta zona de memoria es un componente opcional de la SGA y se utiliza para suministrar un espacio de memoria de gran tamaño utilizado en determinadas circunstancias durante el funcionamiento de la instancia:

- Configuración de servidor compartido.
- Procesos servidor de entrada/salida.
- Buffers paralelos de consulta.
- Operaciones de Backup y recuperación utilizando la utilidad RMAN.

La Large Pool desempeña un papel importante en la realización de acciones de Tuning sobre la base de datos ya que determinadas acciones de localización de memoria para ciertos componentes se llevarían a cabo por la Shared Pool si la Large Pool no estuviera disponible. Debido a posibles grandes requerimientos asociados a las operaciones de entrada/salida y a la utilización de RMAN la Large Pool es un elemento preferente para facilitar su ejecución en lugar de la Shared Pool.

La utilización de la Large Pool facilita a la Shared Pool las operaciones de gestión de sentencias SQL y posibilita evitar la sobrecarga producida por la reducción del espacio de memoria dedicado al manejo de las sentencias SQL.

4.4.3.1.6 Streams Pool

La Streams Pool es una zona de memoria utilizada por los Streams Oracle. Los Streams Oracle son un mecanismo genérico para la compartición de datos. En los Streams Oracle, cada unidad de información compartida se denomina mensaje, y estos mensajes se pueden compartir a través de un flujo. Este flujo puede propagar información dentro del ámbito de una base de datos o de una base de datos a otra. En las rutas de estos Streams se especifica la información y los destinos de la misma.

El resultado de la aplicación de los Streams Oracle es la obtención de una mayor funcionalidad y flexibilidad con respecto a las soluciones tradicionales de captura y de gestión de mensajes, y permitir la compartición de mensajes con otras bases de datos y aplicaciones.

Esta zona de memoria se dimensiona utilizando el parámetro *STREAMS_POOL_SIZE* (0 por defecto).

4.4.3.1.7 Result Cache

La Result Cache es un área contenida en la Shared Pool y permite almacenar los resultados finales de una consulta.

El tamaño máximo de la Result Cache se dimensiona por defecto por el sgbd en función de la cantidad de memoria total disponible para la SGA y del modo de gestión de la memoria utilizada en ese instante. En el caso de necesidad, este tamaño máximo se puede definir por el parámetro *RESULT_CACHE_MAX_SIZE*.

4.4.3.2 Los procesos en segundo plano

Los procesos en segundo plano tienen un papel bien definido en el funcionamiento de la instancia. La mayor parte de los procesos en segundo plano se lanzan durante el inicio de la instancia y se detienen en la parada de la misma. Ciertos procesos pueden ser lanzados y detenidos en el curso del funcionamiento de la instancia. Cada proceso en segundo plano se identifica con un nombre de cuatro caracteres, expresado de la forma *ABCn*, donde *n* es un número o una letra variable para los procesos de los que existen diferentes copias. Los principales procesos en segundo plano son los siguientes:

- DBWn: procesos encargados de escribir los bloques modificados de la Database Buffer Cache en los ficheros de datos.

- LGWR: proceso encargado de escribir la Redo Log Buffer en el fichero de actualización actual.

- CKPT: proceso encargado de registrar un punto de control en la cabecera de los ficheros de datos y en el fichero de control.

- SMON: proceso encargado de llevar a cabo la recuperación de la instancia después de producirse una para anómala.

- ARCn: procesos encargados del almacenamiento de los ficheros de actualización que están llenos.

- CJQn: procesos encargados de ejecutar periódicamente las tareas programadas en el sistema.

- PMON: proceso encargado de la limpieza posterior a una parada anómala de un proceso de usuario.

4.4.3.2.1 DBWn

Los procesos en segundo plano DBWn (Database Writer) están encargados de escribir los bloques modificados de la Database Buffer Cache en el fichero de datos.

Generalmente, una instancia tiene un solo proceso Database Writer referenciado por el nombre DBW0. En los sistemas multiprocesador y multidisco que presentan una actividad paralela de actualización mayor, es posible, incluso aconsejable, iniciar varios procesos Database Writer.

Los procesos DBWn realizan escrituras multibloque, en diferido con respecto a las modificaciones en memoria.

La escritura se desencadena por la ocurrencia de uno de los siguientes eventos:

- Un proceso servidor no localiza espacio disponible en la caché.

- Periódicamente, para hacer avanzar el punto de control (posición en el fichero de actualización a partir de la cual es susceptible comenzar la recuperación de la instancia).

Con respecto a la acción de escritura del proceso DBWn, es importante notar que no hay una sincronización entre la modificación de un bloque en memoria, aunque esté validada (COMMIT), y la escritura en disco de los bloques en cuestión.

4.4.3.2.2 LGWR

El proceso en segundo plano LGWR (Log Writer) se encarga de escribir la Redo Log Buffer en el fichero de actualización actual. LGWR escribe secuencialmente en el fichero de actualización.

La escritura se desencadena por uno de los siguientes eventos:

- Una transacción es validada (COMMIT).

- Database Writer se prepara para escribir los bloques modificados de transacciones no validadas en el fichero de datos.

- El timeout establecido se ha superado (por defecto 3 segundos).

La escritura en el fichero de actualización de la Redo Log Buffer es lo único que se produce durante la validación (COMMIT) de una transacción. Esta operación de escritura es normalmente rápida. Adicionalmente, si el proceso DBWn escribe en el fichero de datos los bloques modificados de transacciones todavía no validadas, la Redo Log Buffer se escribe en el fichero de actualización.

En el caso de una parada anormal de la instancia, el hecho de que un fichero de datos pueda no contener los datos modificados de transacciones validadas o contener los datos modificados de transacciones no validadas, no supone un problema, gracias al proceso de escritura LGWR. Este proceso permite garantizar que los ficheros de actualización contienen la información necesaria para rehacer las transacciones validadas. La posible recuperación de la instancia después de una parada anormal es un proceso automático que no requiere ninguna intervención.

Cuando produce la validación de una transacción, el sgbd de Oracle asocia un número SCN (System Change Number) a la transacción. Este número SCN se registra en el fichero de actualización y puede consultarse en la vista *v$database*. Este número permitirá al sistema determinar en qué instante se encuentra.

4.4.3.2.3 CKPT

Periódicamente, todos los bloques modificados de la Database Buffer Cache se escriben en los ficheros de datos. Aquí es donde interviene el mecanismo de sincronización a través del proceso CKPT (checkpoint). El resultado de la ejecución de este proceso será la armonización de los ficheros de datos y los ficheros de control permitiendo que ambos tipos de ficheros contengan el mismo número de SCN. El proceso CKPT permite garantizar que los bloques modificados en memoria se escriben en los ficheros de datos.

La lógica empleada por el proceso CKPT obliga a que se defina un punto de control en el fichero de actualización (registro del SCN en dicho fichero). Esta posición corresponde a la modificación del bloque más antiguo que no ha sido todavía escrito en el fichero de datos, por lo que este punto marcará el comienzo de los datos a utilizar a la hora de llevar a cabo la recuperación de la instancia.

Uno de los casos en los que se desencadena un punto de sincronización, es el cambio de grupo de ficheros de actualización. En el instante en el que se comienza a utilizar un nuevo grupo, se lleva a cabo la escritura en el fichero de datos de los bloques modificados (no escritos todavía) correspondientes a la información presente en el fichero de actualización.

Los procesos en segundo plano LGWR no pueden comenzar a escribir en un fichero de actualización hasta que la sincronización no haya terminado. Mientras el proceso de sincronización no haya terminado, el fichero de actualización contendrá información que tendría que emplearse en el caso de que fuera necesario llevar a cabo la recuperación de la instancia por producirse una parada anómala.

Las sincronizaciones se producen también por la ocurrencia de los siguientes eventos:

- Parada de la base de datos.
- Tablespace fuera de línea.

El proceso en segundo plano CKPT lleva a cabo el registro del punto de control en la cabecera de los ficheros de datos y en el fichero de control.

4.4.3.2.4 SMON

El proceso en segundo plano SMON (System Monitor) se encarga fundamentalmente de llevar a cabo la recuperación de la instancia después de una parada anómala.

Este proceso lleva a cabo adicionalmente, la liberación de los segmentos temporales inutilizados y compacta el espacio contiguo de los Tablespaces gestionados por el diccionario. Durante la recuperación de la instancia, SMON efectúa dos operaciones:

- Un *roll forward*[65] para aplicar en los ficheros de datos las modificaciones no registradas en transacciones validadas.
- Tras la aplicación de la operación anterior aplica un *rollback*[66] para eliminar de los ficheros de datos las modificaciones registradas de transacciones no validadas.

4.4.3.2.5 PMON

El proceso en segundo plano PMON (Process Monitor) se encarga principalmente de la limpieza posterior aplicada como resultado de una parada anómala de un proceso de usuario.

- Anulación (*rollback*) de las transacciones en curso.
- Liberación de bloqueos y recursos utilizados.

[65] El término *roll forward* se emplea cuando un proceso del sgbd aplica los cambios almacenados en los ficheros de actualización (tanto en línea como almacenados). El SCN de la base de datos se actualiza con la finalidad de reflejar la aplicación de estos cambios. Esta operación se produce durante la recuperación de una base de datos, un Tablespace o un fichero de datos.
[66] La acción de *rollback* es el proceso que se aplica para deshacer una transacción que no ha sido validada (acción de commit).

El proceso en segundo plano PMON es capaz de detectar situaciones en las que un proceso de usuario que ha abierto una sesión en el servidor, ya no está "presente" y no ha cerrado la sesión. La causa de la no "presencia" del proceso de usuario es variable: fin anómalo de la aplicación en el puesto de usuario, corte de red, etc.

En cualquier caso, el proceso PMON se encarga de la limpieza efectuando una anulación (*rollback*) de la transacción. Esta anulación deja libre los bloqueos que tiene la transacción. Desde un punto de vista de integridad de datos, la desaparición de procesos de usuario no supone ningún problema tras la intervención de este proceso.

4.4.3.2.6 CJQn

Los procesos en segundo plano CJQn (Job Queue) se encargan de ejecutar periódicamente las tareas programadas por el sistema.

Un proceso coordinador (CJQ0) supervisa si hay trabajos pendientes de ejecución. Si localiza un trabajo pendiente, un proceso denominado "esclavo" y que se reconoce mediante el nombre (J000…J999) se encarga de la ejecución de dicho trabajo.

4.4.3.2.7 ARCn

Los procesos en segundo plano ARCn (*Archiver*) se encargan del almacenamiento de los ficheros de actualización que están llenos. Para que este proceso esté activo la base de datos debe encontrarse en modo *ARCHIVE LOG*.

4.4.3.3 Los procesos servidor

Los procesos de servidor son los encargados del tratamiento de las consultas de usuario y fundamentalmente, de cargar los datos necesarios de la Database Buffer Cache. Estos procesos servidor se comunican (localmente o a través de la red) con un proceso de usuario correspondiente a la aplicación que esté utilizando.

En la configuración por defecto, el sgbd de Oracle lanza un proceso servidor dedicado para cada usuario (*dedicated server configuration*). Este proceso únicamente trata las peticiones realizadas por el usuario en cuestión.

Si es necesario, se puede configurar el sgbd de Oracle como servidor compartido (shared server configuration) con la finalidad de tener los procesos servidor compartidos por varios procesos de usuario. En la configuración de servidor compartido, únicamente un pequeño número de procesos servidor se comparten entre los diferentes procesos de usuario y pueden tratar de forma indistinta las consultas de cualquier usuario.

Esta configuración permite limitar el número de procesos ejecutados en el servidor y optimizar la utilización de los recursos del sistema.

En ambos casos, la comunicación entre el proceso de usuario y el proceso servidor, es local, si la aplicación se ejecuta en el propio servidor o bien se efectúa a través de la red (Oracle Net) si la aplicación se ejecuta desde un puesto diferente.

4.4.3.4 PGA (Program Global Area)

La PGA (Program Global Area) es la zona de memoria privada asignada a los diferentes procesos. En un proceso servidor, la PGA contiene:

- Una zona de trabajo SQL (*SQL Work Area*) asignada dinámicamente para ciertas operaciones (ordenación, por ejemplo).
- Información de la sesión.
- Información del tratamiento de consultas de la sesión.
- Variables de sesión.

La zona de memoria asociada a todos los procesos servidor se conoce con el nombre de PGA agregada (aggregated PGA) o PGA de la instancia (instance PGA).

En una configuración de servidor compartido, una parte de la PGA está, de hecho, almacenada en la SGA, en la Large Pool o, por defecto, en la Shared Pool. En este tipo de configuración, no es forzosamente el mismo proceso servidor el que va a procesar dos consultas sucesivas de la memoria de procesos de usuario. La información relativa a esta sesión de usuario debe ser, por tanto accesible al conjunto de procesos servidor. Esta es la razón por la que, en una configuración de servidor compartido, una parte de la PGA de los procesos servidor se almacena en la SGA. Cuando un proceso servidor está tratando una consulta de un proceso de usuario, recarga el contexto del proceso usuario a partir de la SGA. Por estos motivos, en una configuración de servidor compartido, es necesario aumentar el espacio de la SGA (posiblemente habilitando la Large Pool) aunque las necesidades globales de memoria sean similares (ya que los procesos servidor como tales, utilizarán menos memoria).

En versiones anteriores al sgbd Oracle 9i, el tamaño de la zona de trabajo SQL se controlaba por medio de varios parámetros:

- *SORT_AREA_SIZE*: Espacio destinado a las operaciones de ordenación.
- *HASH_AREA_SIZE*: Parámetro relativo a la utilización de Hash Joins[67].

67 Este método es usado para enlazar grandes conjuntos de datos, en los que la condición de join es de igualdad. El optimizador usa la más pequeña de las dos tablas fuentes para construir una tabla hash sobre la clave de join, en memoria. Cuando se recuperan los registros de la tabla más grande, se examina la tabla hash para encontrar las filas enlazadas en la otra tabla.

- *BITMAP_MERGE_AREA_SIZE*: Parámetro asociado a la fusión de mapas de bits e índices de mapas de bits.

- *CREATE_BIT_MAP_AREA_SIZE*: Parámetro relativo a la creación de índices de mapas de bits.

A partir de la versión del sgbd Oracle 9i, se permite la gestión automática y global de la PGA agregada de los procesos servidor. En este caso, es suficiente con definir la cantidad total de memoria total que la PGA agregada puede utilizar y dejar que el propio sgbd reparta la memoria entre los diferentes procesos en función de las necesidades. En este modo de funcionamiento, los parámetros presentados anteriormente son ignorados. El tamaño de la PGA agregada de los procesos servidor se define por el parámetro *PGA_AGGREGATE_TARGET*.

En la versión 11g del sgbd de Oracle, la PGA puede también ser gestionada automáticamente en el seno de la memoria total de la instancia.

4.4.3.5 El fichero de parámetros

Durante su inicio, la instancia lleva a cabo la lectura de un fichero de parámetros que contiene los parámetros de inicialización. Estos parámetros de inicialización permiten a la instancia asignar la memoria deseada a las diferentes estructuras de la SGA, y encontrar el número y emplazamiento de los ficheros de control de la base de datos a abrir. Este fichero es gestionado por el administrador de base de datos (DBA).

Históricamente, el sgbd de Oracle ha utilizado un fichero de parámetros de tipo texto para el inicio de la instancia de la base de datos. A partir de la versión 9i, es posible utilizar un fichero de parámetros binario almacenado en el servidor (server parameter file SPFILE). Podría considerarse este fichero como un repositorio centralizado de parámetros de inicialización.

El fichero de parámetros del servidor es un fichero binario que puede ser generado a partir de un fichero de texto de parámetros en el que cada uno de los parámetros se especifican de la forma *nombre_parámetro=valor*. A priori, todos los parámetros son opcionales y tienen un valor por defecto. En este tipo de ficheros se pueden incluir comentarios precedidos por el carácter #. Los valores de cada uno de los parámetros se pueden especificar entre comillas dobles si contienen algún tipo de carácter especial (igual, espacios en blanco…). Los valores múltiples se pueden especificar entre paréntesis, separados por comas.

```
###########################################################################
# Example INIT.ORA file
## This file is provided by Oracle Corporation to help you start by providing
# a starting point to customize your RDBMS installation for your site.
#
# NOTE: The values that are used in this file are only intended to be used
# as a starting point. You may want to adjust/tune those values to your
# specific hardware and needs. You may also consider using Database
# Configuration Assistant tool (DBCA) to create INIT file and to size your
# initial set of tablespaces based on the user input.
###########################################################################

# Change '<ORACLE_BASE>' to point to the oracle base (the one you specify at
# install time)

db_name='ORCL'
memory_target=1G
processes = 150
audit_file_dest='<ORACLE_BASE>/admin/orcl/adump'
audit_trail ='db'
db_block_size=8192
db_domain=''
db_recovery_file_dest='<ORACLE_BASE>/flash_recovery_area'
db_recovery_file_dest_size=2G
diagnostic_dest='<ORACLE_BASE>'
dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB)'
open_cursors=300
remote_login_passwordfile='EXCLUSIVE'
undo_tablespace='UNDOTBS1'
# You may want to ensure that control files are created on separate physical
# devices
control_files = (ora_control1, ora_control2)
compatible ='11.1.0'
```

Figura 12.Ejemplo de fichero de configuración de parámetros.

4.4.4 El Administrador de bases de datos

El administrador de base de datos es la persona o conjunto de personas encargada/s fundamentalmente de llevar a cabo las siguientes tareas:

- Instalación de los productos.
- Creación/inicio/parada de base de datos.
- Gestión de las estructuras de almacenamiento.
- Gestión de usuarios y privilegios.
- Copias de seguridad y restauración.

4.4.4.1 Cuentas del sgbd Oracle de administración

Después de su creación, una base de datos Oracle contiene siempre dos cuentas con todos los derechos de administración:

- *SYS*: contraseña por defecto: *change_on_install*.
- *SYSTEM*: contraseña por defecto: *manager*.

Desde la aparición del sgbd Oracle 9i release 2, las contraseñas por defecto de estos usuarios se pueden cambiar en el instante de la creación de la base de datos.

SYS es el propietario del diccionario de datos; *SYSTEM* puede ser el propietario de las tablas utilizadas por las herramientas Oracle.

Un privilegio complementario particular (*SYSDBA* o *SYSOPER*) es necesario para ciertas operaciones (inicio, parada, etc.). Además, la activación de estos privilegios *SYSDBA* o *SYSOPER* necesita un mecanismo de autenticación particular, debido a que la base de datos puede dejar de estar disponible. Esta autenticación es posible realizarla a través del sistema operativo o a través de la utilización de un fichero de contraseñas.

4.4.4.2 Identificación privilegiada SYSDBA y SYSOPER

El privilegio *SYSDBA* permite realizar todas las operaciones "pesadas" de administración, fundamentalmente la creación de una base de datos, las paradas e inicios, la creación de un fichero de parámetros servidor, las recuperaciones, etc. Da acceso a todos los datos de la base de datos. La conexión se efectúa implícitamente en el esquema de *SYS*.

El privilegio *SYSOPER* ofrece prácticamente los mismos derechos que *SYSDBA*, con la excepción notable de la creación de la base de datos. El acceso está restringido únicamente a los datos del diccionario de datos. La conexión se efectúa implícitamente en el esquema *PUBLIC*.

Desde la versión del sgbd Oracle 9i no es posible la conexión utilizando la cuenta *SYS* sin privilegios *SYSDBA*.

El proceso de administración habitual no necesita el privilegio *SYSDBA* o *SYSOPER*; habitualmente es suficiente con utilizar la cuenta *SYSTEM* que permite llevar a cabo las siguientes acciones:

- Gestión de estructuras de almacenamiento.
- Gestión de usuarios.
- Gestión de esquemas.

El privilegio *SYSDBA* es necesario para llevar a cabo las siguientes acciones:

- El inicio y la parada de la base de datos.
- La creación de una base de datos.
- Las operaciones de recuperación de una base de datos.

En versiones anteriores del sgbd de Oracle, era posible utilizar *CONNECT INTERNAL* con la finalidad de obtener estos privilegios particulares. Este tipo de conexión no está disponible a partir de la versión 9i. A partir de esta versión habrá que llevar a cabo una conexión *AS SYSDBA* (equivalente a efectos de privilegios).

La autenticación *SYSDBA* o *SYSOPER* por parte del sistema operativo, no está disponible para las conexiones llevadas a cabo utilizando una red (salvo en el caso de que se utilice una red segura). En este caso será necesario utilizar un método de autenticación basado en un fichero de contraseñas.

Para la autenticación local de la base de datos se puede utilizar tanto una autenticación de sistema operativo o una autenticación mediante un fichero de contraseñas. En el primer caso, se debe asegurar que los grupos y cuentas correspondientes del sistema operativo están adecuadamente protegidos. En el segundo caso, se debe asegurar que el fichero de contraseñas y el usuario *orapwd* están bien protegidos.

4.4.4.3 Otras cuentas Oracle

En el instante de creación de una base de datos, es posible crear adicionalmente otras cuentas Oracle. Entre estas cuentas destacan fundamentalmente *SYSMAN* y *DBSNMP*.

SYSMAN: es una cuenta que puede ser utilizada para llevar a cabo labores de administración en Oracle Enterprise Manager. *SYSMAN* es una cuenta DBA.

DBSNMP: es una cuenta utilizada por el agente Oracle Enterprise Manager para la supervisión y la gestión de una base de datos.

4.4.5 El diccionario de datos

El diccionario de datos está constituido por una serie de tablas y vistas que permiten obtener información sobre el contenido de una base de datos:

- Las estructuras de almacenamiento.
- Los usuarios y sus derechos.
- Los objetos (tablas, vistas, índices, procedimientos, funciones, etc).

El diccionario de datos pertenece al usuario *SYS* y se almacena en el Tablespace *SYSTEM*. Se crea en el instante de la creación de la base de datos y es actualizado automáticamente por el sgbd cuando se ejecutan sentencias SQL asociadas al DDL (CREATE, ALTER, DROP).

El diccionario de datos se carga en memoria en la zona Dictionary Cache contenida en la Shared Pool y el sgbd Oracle lo utiliza para llevar a cabo el tratamiento de las consultas realizadas por las aplicaciones/usuarios.

Hay dos grupos de tablas/vistas en el diccionario de datos.

- Las tablas y las vistas estáticas: estas tablas y vistas están basadas en las verdaderas tablas almacenadas en el Tablespace *SYSTEM*. Son accesibles únicamente cuando la base de datos está "abierta".

- Las tablas y las vistas dinámicas de rendimiento: estas tablas y vistas no están basadas en verdaderas tablas sino en la información en memoria o en extractos del fichero de control. No obstante, son accesibles como si fueran tablas/vistas reales y dan información sobre el funcionamiento de la base de datos (fundamentalmente sobre el rendimiento). La mayoría son accesibles aunque la base de datos no se encuentre en el estado "abierto".

4.4.5.1 Las vistas estáticas

Hay tres grandes grupos de vistas estáticas, donde cada grupo se diferencia por su prefijo[68]:

- *USER_%*[69]: información sobre los objetos que pertenecen al usuario.

- *ALL_%*: información sobre los objetos a los cuales el usuario tiene acceso (los suyos más los objetos sobre los que se le han otorgado derechos de acceso).

- *DBA_%*: información sobre los objetos de la base de datos.

Estas tres categorías permiten filtrar la información del diccionario de datos en relación a los derechos de los usuarios. La información accesible en las vistas *USER_* constituye un subconjunto de la información accesible en las vistas *ALL_* que, al mismo tiempo, constituye un subconjunto de la información accesible en las vistas *DBA_*.

[68] Detrás del prefijo, el resto del nombre de la vista representa la información accesible.
[69] El carácter % es utilizado como comodín para representar una secuencia de caracteres.

En las vistas de la categoría *ALL_* y *DBA_* que hacen referencia a los objetos de los esquemas, la columna *OWNER* permite conocer el propietario del objeto.

Las vistas referenciadas en la siguiente tabla son particularmente útiles para la descripción de un esquema:

Vista	Descripción
%_OBJECT	*Información del objeto, como el estado (status) o el instante de aplicación de la última sentencia DDL.*
%_TABLES	*Información de las tablas, como el Tablespace asociado, parámetros de almacenamiento y número de filas.*
%_TAB_COLUMNS	*Información sobre las columnas que integran tanto las tablas como las vistas, incluyendo su tipo de datos.*
%_INDEXES	*Información relativa a los índices. Tipo, unicidad y tabla a la cual se referencia.*
%_IND_COLUMNS	*Información de las columnas que forman parte de los índices, incluyendo su orden dentro del índice.*
%_TRIGGERS	*Información sobre los triggers definidos en la base de datos, como el tipo, evento y cuerpo del trigger*[70].
%_CONSTRAINTS	*Información relativa a las restricciones establecidas en la base de datos.*
%_CONS_ COLUMNS	*Información asociada a las columnas sobre la que se aplican las restricciones definidas en la base de datos.*
%_VIEWS	*Información de las vistas, incluyendo la definición de la vista.*
%_SYNONYMS	*Información del sinónimo, como el objeto referenciado y el database link*[71].
%_SEQUENCES	*Información de la sequence, como caché, cycle y último número.*
%_SOURCE	*Código fuente almacenado (excepto triggers).*

Tabla 3. Vistas del diccionario de datos relevantes en la descripción del esquema.

[70] Un trigger (o disparador) en una base de datos , es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación.

[71] Un database link (DBlink) constituye una definición de cómo establecer una conexión de una base de datos Oracle a otra.

El sgbd de Oracle permite la utilización de sinónimos sobre algunas vistas:

Vista	Sinónimo
USER_TAB_COLUMNS	*COLS*
DICTIONARY	*DICT*
USER_INDEXES	*IND*
USER_OBJECTS	*OBJ*
USER_SEQUENCES	*SEQ*
USER_SYNONYMS	*SYN*
USER_TABLES	*TABS*

Tabla 4. Vistas del diccionario de datos y sinónimos asociados.

Hay una serie de vistas asociadas al diccionario de datos que permiten consultar la descripción de todas las tablas y vistas del diccionario de datos:

- *DICTIONARY*: esta vista es muy práctica para recuperar los nombres de las vistas que tienen que ver con determinados elementos clave de la base de datos.

- *DICT_COLUMNS*: esta vista permitirá obtener una descripción completa de las columnas asociadas a un determinado objeto de la base de datos.

4.4.5.2 Las vistas dinámicas de rendimiento (*V$*)

Las vistas dinámicas de rendimiento utilizan el prefijo *V$*. Detrás del prefijo, el resto del nombre de la vista hace referencia a la información accesible.

Salvo excepción, estas vistas sólo son accesibles por el DBA.

Algunos ejemplos de vistas dinámicas utilizadas habitualmente se exponen en la siguiente tabla:

Vista	Contenido
V$INSTANCE	Información de la instancia.
V$DATABASE	Información de la base de datos.
V$SGA	Información de la SGA.
V$SGAINFO	Información sobre el tamaño de todos los elementos que componen la SGA.
V$PARAMETER[72]	Información de los parámetros utilizados en la instancia.

Tabla 5.Vistas dinámicas de rendimiento utilizadas habitualmente.

[72] La vista *V$PARAMETER* es una de las pocas vistas del diccionario de datos que almacena información en minúsculas.

Las vistas dinámicas de rendimiento están también descritas en las vistas *DICTIONARY* y *DICT_COLUMNS*. Como complemento a estas vistas se puede utilizar la vista *V$FIXED* para obtener información sobre la definición interna de las vistas dinámicas.

Capítulo 5

La Metodología Ágil Scrum

5.1 Introducción

En este capítulo se procederá a realizar una exposición de la metodología ágil Scrum que será la que se utilizará en el desarrollo de la aplicación *AAS11*. Durante este capítulo se procederá a describir los orígenes históricos, los fundamentos de la metodología, las fases en las que se descompone, el equipo de trabajo definido y las distintas herramientas y diagramas que se utilizan habitualmente durante su aplicación.

5.2 Orígenes de la metodología

En el año 1986, Irotaka Takeuchi e Ikujiro Nonaka publicaron un artículo titulado "*El Nuevo Juego del desarrollo de Nuevos Productos*" [TN86]. En este artículo los autores narran sus observaciones dentro de diversas empresas relacionadas con el mundo de la tecnología como Xerox, NEC, Canon y Honda, y describen la forma en la que las empresas más innovadoras, en el seno de mercados sujetos a una rápida evolución, estaban enfocando los retos de la adaptación al cambio y de los ciclos de desarrollo cada vez más cortos. Con la finalidad de describir este enfoque, que calificaron como holístico[73], recurrieron a una analogía con el juego del Rugby, donde la pelota no progresa de forma secuencial desde la defensa a los delanteros, sino que todo el equipo participa simultáneamente en la jugada mediante formaciones características de este deporte, como la melé, que en inglés se denomina scrum.

[73] El término holístico hace referencia a una concepción basada en la integración total frente a un determinado concepto o situación particular.

Aunque el artículo de Takeuchi y Nonaka no estaba enfocado al desarrollo de software, ha tenido bastante influencia sobre determinadas metodologías dentro de este ámbito, y en concreto sobre Scrum. Una de las ideas que propone este artículo es reemplazar el proceso clásico de ingeniería secuencial, donde cada fase es ejecutada de forma sucesiva y por equipos distintos de personas, por un proceso adaptativo, apoyado sobre la prueba y error, donde las fases se solapan y donde es un mismo equipo multifuncional el que ejecuta todo el proceso de desarrollo. El objetivo es conseguir responder al desarrollo de nuevos productos no sólo atendiendo a criterios de calidad, costes y diferenciación sino también considerando criterios como la flexibilidad y la rapidez.

La primera referencia al término Scrum en el contexto de desarrollo de software corresponde al libro *Wicked Problems, Righteus Solutions* de Peter DeGrace y Leslie Hulet Stahl, editado en 1990 [DH90]. Esta obra describe una determinada clase de problemas cuya solución es difícilmente alcanzable mediante los enfoques predictivos tradicionales. El término utilizado para referirse a esta clase de problemas es el de problemas "perversos", propuesto por el alemán Horst Rittel en 1973 [RW73]. Los problemas perversos no disponen de una formulación definitiva, nunca son solucionados en todos sus aspectos, y sus soluciones no son ciertas o falsas sino mejores o peores. Generalmente este tipo de problemas están asociados a otros problemas, y dependiendo del enfoque seleccionado tienen soluciones muy distintas. Adicionalmente, las condiciones a las que están sometidos estos problemas y los recursos disponibles para resolverlos cambian con el tiempo, por lo que se requieren soluciones adaptativas.

En 1995, Ken Schwaber y Jeff Sutherland presentaron en una conferencia la metodología que hoy se conoce como Scrum. En 2001 Ken Schwaber y Mike Beedle detallaron la metodología en su libro Agile Software Development with Scrum [SB01]. En líneas generales, Scrum puede considerarse como un proceso de prototipado basado en iteraciones cortas y frecuentes dirigido por la necesidad de negocio en orden de importancia.

5.3 Fundamentos de la metodología

Scrum es un marco ágil para el desarrollo de productos. El desarrollo de productos generalmente se estructura alrededor de proyectos, por lo que se puede también pensar en Scrum como un marco para la gestión ágil de proyectos.

En lugar de ser una metodología completa, Scrum está constituido por un esqueleto simple basado en principios, prácticas y valores ágiles. Esto significa que en lugar de suministrar descripciones completas y detalladas acerca de cómo todo proceso debe ser realizado dentro del marco de un proyecto, Scrum es ligero y prácticamente todos los detalles de implementación se dejan para su resolución al equipo que está desarrollando el trabajo.

En Scrum, los proyectos se descomponen temporalmente en pequeños periodos de duración comprendida entre dos y cuatro semanas. Cada uno de estos periodos se denomina Sprint[74]. Tras la finalización de cada Sprint el equipo se compromete a realizar una entrega del producto con un conjunto cerrado de funcionalidades que han sido definidas para ese periodo.

El siguiente Sprint comienza inmediatamente tras finalizar el anterior. Los Sprints tienen tiempo fijo (terminan en una fecha previamente establecida se haya completado o no el trabajo). Los Sprints, por definición, no pueden ser extendidos nunca.

Al principio de cada Sprint, el equipo encargado de la aplicación de la metodología elige una serie de requisitos de usuario final (generalmente denominado historias de usuario o para abreviar, simplemente historias) de una lista priorizada y ordenada denominada Product Backlog. El equipo se compromete a dedicar todos sus esfuerzos con la finalidad de completar estas historias de usuario. Durante el Sprint, las historias de usuario no pueden ser modificadas. Cada día que compone el Sprint el equipo se reúne para inspeccionar el progreso y ajustar los próximos pasos a llevar a cabo para completar el trabajo pendiente. Al final de cada Sprint, el equipo repasa el trabajo realizado con las diferentes partes interesadas en el proyecto y expone las historias de usuario que ha completado con éxito.

5.4 Roles en Scrum

En Scrum existen tres roles: el Product Owner, el equipo y el Scrum Master. Cada uno de estos roles se describe de forma detallada en los siguientes puntos.

5.4.1 Product Owner

El Product Owner (dueño del producto) es el responsable de garantizar el máximo retorno de la inversión del proyecto. Para realizar esto, identifica funcionalidades que desea incorporar al producto, especificándolas en una lista priorizada y ordenada. Debe decidir qué funcionalidades deben estar en la parte superior de la lista para cada Sprint y debe realizar de forma sucesiva una labor de actualización, refinamiento y priorización de la lista de requisitos a medida que el proyecto evoluciona.

La priorización de la lista puede verse influenciada por diversos factores como dependencias técnicas, necesidad de satisfacer requisitos clave, mitigación de riesgos durante la elaboración del producto…

[74] Término que en inglés se utiliza para designar una carrera corta.

En algunos casos el Product Owner y el cliente final son la misma persona. Esta circunstancia es típica en desarrollos internos. En la mayoría de casos este rol estará desempeñado por una persona ajena a la organización. Es una condición fundamental que este rol esté desempeñado por una única persona. Este hecho es fundamental para evitar conflictos de prioridades y ambigüedades en la comunicación con el equipo.

5.4.2 El equipo

El equipo de desarrollo será el encargado de construir el producto que el Product Owner ha descrito. Este equipo se caracteriza por ser multifuncional (incluye todos los roles y conocimientos necesarios para realizar un desarrollo completo del producto) y es auto gestionado, con un alto grado de autonomía y responsabilidad. El equipo decide a qué comprometerse, y cuál es la mejor forma de alcanzar el objetivo sobre el que se ha establecido dicho compromiso.

El equipo de desarrollo diseña, desarrolla, prueba el producto y ayuda al Product Owner con ideas con respecto a la construcción del producto. El equipo también será el responsable de supervisar la calidad interna del producto.

Con respecto a la organización del equipo de desarrollo, se puede destacar que rompe con la tradicional pirámide jerárquica. La división entre análisis, diseño y construcción es reemplazada por un enfoque global, donde todos los desarrolladores analizan, diseñan, documentan, prueban y construyen. Con este tipo de organización ninguna actividad es secuencial y todas confluyen en un repositorio común de código como se muestra en la siguiente figura.

Figura 13.Enfoque global del desarrollo.

El reparto de las tareas se realiza por funcionalidades, componentes o subsistemas y todas las personas participan en todas las actividades. Como se ha comentado anteriormente, este hecho requiere de individuos polivalentes y auto-organizados, entre los que se fomenta la responsabilidad asumida. Al repartir las actividades de esta forma, lógicamente no son estrictamente necesarios los documentos de análisis y diseño como nexo entre unas etapas y otras, típicos de un proceso de desarrollo en cascada. El analista no necesita entregar al programador un documento de diseño, ni un plan de pruebas, ya que el analista, programador y encargado de pruebas es una misma persona o un mismo equipo. Esta documentación puede escribirse si se considera que de esta forma se puede llegar a ser más productivo, pero, a priori, la única documentación necesaria es la documentación final que permitirá a terceros utilizar el producto, es decir, el manual de usuario.

El tamaño óptimo de los equipos de desarrollo es entre cinco y nueve individuos. Un tamaño superior comienza a ser más complicado de gestionar, ya que resulta incómodo reunirse, más complicado entenderse y alcanzar acuerdos sobre decisiones. Cuando un proyecto es particularmente grande y requiere de más participantes, lo que debe hacerse es descomponer el proyecto en subproyectos, gestionados por distintos equipos. No es específico de Scrum, pero cuando un proyecto es mantenido por una comunidad amplia de desarrolladores, ya sea dentro o más allá de las fronteras de una organización, lo normal es que exista un pequeño grupo de no más de nueve personas que lo mantengan o controlen directamente, y que coordinen las aportaciones del resto.

5.4.3 El Scrum Master

El Scrum Master, o encargado, es una persona del equipo que sirve como interlocutor para resolver las dificultades que puedan aparecer, y que se encarga de asegurar las mejores circunstancias posibles de trabajo, liberando al resto de posibles distracciones o interrupciones. No tiene por qué ser un líder de equipo, ni corresponde con el rol de jefe de proyecto de la ingeniería tradicional. Con Scrum no es necesario ningún jefe de proyecto en el sentido formal porque cada persona ya sabe lo que tiene que hacer. Las personas que requieren que alguien les diga constantemente lo que tienen que hacer, o que supervisen su trabajo, necesitarán un periodo de adaptación para ser capaces de asumir responsabilidades y poder trabajar en equipos auto-organizados.

5.5 Descripción de la metodología

Con la finalidad de aplicar la metodología Scrum se deben llevar a cabo los siguientes pasos:

Inicialmente, el Product Owner debe articular la visión del producto a desarrollar. Esta visión será transformada en una lista priorizada, ordenada y estimada de funcionalidades llamada Product Backlog. En esta lista se definirá el alcance del proyecto y la planificación del mismo. Esto debe realizarse antes de comenzar el primer Sprint, durante un periodo que habitualmente se denomina Sprint Cero.

Un buen Product Backlog está constituido por una lista priorizada de requerimientos (historias de usuario), expresadas en términos simples como funcionalidades requeridas por el usuario final. Únicamente existe un Product Backlog y cada historia de usuario contenida en el mismo tiene una prioridad única. Este hecho obliga al Product Owner a tomar decisiones sobre todo el espectro de demandas del proyecto. El Product Backlog puede ser actualizado en cualquier momento por el Product Owner para reflejar cambios en las necesidades del cliente, nuevas ideas, cambios en los requisitos derivados de problemas técnicos…

Una vez realizada una primera definición del Product Backlog éste debe ser estimado. El equipo es el responsable de llevar a cabo las estimaciones de las historias de usuario.

Teniendo en cuenta factores como el tamaño, valor de negocio, y otras variables importantes como pueden ser el riesgo y las dependencias existentes, el Product Owner priorizará el Product Backlog con la ayuda y el asesoramiento del equipo.

El siguiente paso será llevar a cabo la planificación de una Release[75]. Los equipos de Scrum enfocan su planificación únicamente alrededor de la siguiente Release a llevar a cabo. La finalidad, es obtener una planificación realista.

La planificación de Release en Scrum consiste en identificar el conjunto de funcionalidades que se desea incluir en la próxima Release, dividirlas en elementos suficientemente pequeños estimados y priorizados, y definir la fecha de lanzamiento deseada. Una de las herramientas que se utilizarán en este instante será un gráfico Burn-Down que se utilizará para mostrar de forma visual el alcance de cada uno de los Sprints para llevar a cabo los objetivos. Este gráfico se crea durante la planificación de la Release y es actualizado Sprint tras Sprint por el Product Owner.

[75] El término Release hace referencia a una versión del producto tratado. Una Release identifica el instante en el que una aplicación es puesta en producción o es lanzada al mercado.

Al comienzo de cada Sprint, tiene lugar la reunión de planificación del Sprint. Durante la primera parte de la reunión, el Product Owner y el equipo (con la ayuda del Scrum Master) analizan la funcionalidad de alta prioridad que el Product Owner está interesado en ver implementada durante el Sprint que comienza. El equipo seleccionará tantos elementos de la parte superior del Product Backlog como considere, para llevarlos a cabo en el siguiente Sprint. Será el propio equipo quien decide cuánto trabajo será realizado, no el Product Owner. La definición de la metodología es muy clara en este sentido y el objetivo es lograr un compromiso del equipo con respecto al trabajo a desarrollar. Si la cantidad de trabajo no alcanza las expectativas del Product Owner, se considera como problema de planificación o capacidad pero en ningún caso el Product Owner puede imponer requerimientos al equipo, únicamente decidir la prioridad de los mismos.

Durante la segunda parte de la reunión de planificación, el equipo se dedicará a estudiar las historias de usuario que se han seleccionado, generalmente realizando un análisis y diseño de las mismas. A medida que esto se lleva a cabo, las historias de usuario se descomponen en tareas, que se definen como el conjunto de pasos necesario para completar cada una de las historias de usuario. Habitualmente las tareas son de naturaleza técnica, y no necesariamente deben ser comprendidas por el Product Owner. Estas tareas deben ser relativamente pequeñas, intentando no superar cada una de ellas un día de trabajo.

A medida que el equipo descompone las historias en tareas, asume un conocimiento más detallado de las mismas. En este instante, el equipo podría percatarse de que ha intentado asumir mayor cantidad de trabajo de la realizable pero estaría a tiempo de eliminar alcances del Sprint. Una vez que se han confirmado, las historias de usuario seleccionadas y descompuestas en tareas quedan comprometidas para el Sprint y no pueden eliminarse del mismo. Este conjunto de historias de usuario constituyen la Pila del Sprint (Sprint Backlog). Cualquier nueva funcionalidad o cambio que se desee abordar debe aplazarse hasta el siguiente Sprint.

5.6 Los Sprints y las reuniones

En Scrum los proyectos se descomponen temporalmente en pequeños periodos de duración comprendida entre dos y cuatro semanas. Cada uno de estos periodos se denomina Sprint. Tras la finalización de cada Sprint, el equipo se compromete a realizar una entrega del producto con un conjunto cerrado de funcionalidades para este periodo. Durante este ciclo se celebran una serie de reuniones que se exponen en las siguientes secciones.

5.6.1 Reunión de planificación del Sprint

El primer día del Sprint se dedica a planificar el Sprint Backlog, que está compuesto del conjunto de funcionalidades o características que serán implementadas en dicho periodo. La reunión puede dividirse en dos partes:

- En primer lugar, el Product Owner prioriza una lista de requisitos o funcionalidades de las que le gustaría disponer al final del periodo.
- En segundo lugar, el equipo de desarrollo analiza qué tareas serían necesarias, y estima cuánto trabajo podrían completar durante ese ciclo, repartiéndose las tareas de forma conjunta.

5.6.2 Reunión de Scrum

Diariamente se realiza una reunión de seguimiento que se denomina Scrum, de no más de quince minutos, siempre en el mismo lugar y a la misma hora. En dicha reunión cada miembro responde a tres preguntas:

- Qué ha hecho desde la última reunión.
- Qué piensa hacer en adelante.
- Qué dificultades piensa que pueden impedirle cumplir sus objetivos.

5.6.3 Reunión de revisión y retrospectiva

Al final de cada periodo hay una tercera reunión de revisión y retrospectiva, que tiene dos partes:

- La primera parte consiste en una demostración al cliente de las nuevas funcionalidades completadas.
- La segunda parte tiene carácter retrospectivo, y en ella se analizan aquellas dificultades que el equipo ha encontrado durante el periodo, y cómo podrían mejorar.

5.7 Diagramas y Herramientas

Scrum se apoya sobre un conjunto de artefactos o diagramas, descritos a continuación, que pueden ser mantenidos mediante una simple hoja de cálculo.

5.7.1 El Product Backlog

La lista de objetivos/requisitos priorizada representa la visión y expectativas del cliente respecto a los objetivos y entregas del producto o proyecto. El cliente es el responsable de crear y gestionar la lista. Dado que debe reflejar las expectativas del cliente, esta lista permite involucrarle en la dirección de los resultados del producto o proyecto.

- Contiene los objetivos/requisitos de alto nivel del producto o proyecto, que se suelen expresar en forma de historias de usuario. Para cada objetivo/requisito se indica el valor que aporta al cliente y el coste estimado de completarlo. La lista está priorizada balanceando el valor que cada requisito aporta al negocio frente al coste estimado que tiene su desarrollo, es decir, basándose en el Retorno de la Inversión (ROI).

- En la lista se indican las posibles iteraciones y las entregas (Releases) esperadas por el cliente (los puntos en los cuales desea que se le entreguen los objetivos/requisitos completados hasta ese momento), en función de la velocidad de desarrollo del (los) equipo(s) que trabajará(n) en el proyecto. Es conveniente que el contenido de cada iteración tenga una coherencia, de manera que se reduzca el esfuerzo de completar todos sus objetivos.

- La lista también tiene que considerar los riesgos del proyecto e incluir los requisitos o tareas necesarios para mitigarlos.

Antes de iniciar la primera iteración, el cliente debe tener definida la meta del producto o proyecto y la lista de requisitos creada. No es necesario que la lista sea completa ni que todos los requisitos estén detallados al mismo nivel. Basta con que estén identificados y con suficiente detalle los requisitos más prioritarios con los que el equipo empezará a trabajar. Los requisitos de iteraciones futuras pueden ser mucho más amplios y generales. La incertidumbre y complejidad propia de un proyecto hacen conveniente no detallar todos los requisitos hasta que su desarrollo esté próximo. De esta manera, el esfuerzo de recoger, detallar y desarrollar el resto de requisitos (menos prioritarios) está repartido en el período de ejecución del proyecto.

En el caso del desarrollo de un producto, la lista va evolucionando durante toda la vida del producto (los requisitos "emergen"). En el caso de un proyecto, conforme éste avance, irán apareciendo los requisitos menos prioritarios que falten. Esto produce varias ventajas:

- Se evita caer en parálisis de análisis al inicio del proyecto, de manera que se puede iniciar antes el desarrollo y el cliente puede empezar a obtener resultados útiles.

- Se evita analizar en detalle requisitos no prioritarios que podrían cambiar durante el transcurso del proyecto, dado que el cliente conocerá mejor cuál ha de ser el resultado a conseguir, o bien por que podrían ser reemplazados por otros.

- Puede llegar a un punto del proyecto en que no valga la pena analizar ni desarrollar los requisitos restantes, dado el poco retorno de inversión (ROI) que tienen.

En la siguiente tabla se puede ver un ejemplo básico de Product Backlog:

ID	Descripción	Prioridad	Coste	Completado	Sprint 1	2	3	4	5
1	Registro de alta	0.9	5	100	70	30	0	0	0
2	Notificaciones por email	0.1	2	100	30	40	30	0	0
3	Provisión del servicio	0.8	10	50	0	0	50	0	0
4	Renovación mensual	0.5	4	0	0	0	0	0	0
5	Baja de servicio	0.8	5	0	0	0	0	0	0
6	Modificación de datos	0.8	5	0	0	0	0	0	0
7	Consulta de actuaciones	0.1	2	0	0	0	0	0	0

Tabla 6.Ejemplo básico de Product Backlog.

5.7.2 El Sprint Backlog

Una vez descompuesto y priorizado el alcance total, en cada periodo se realiza una fracción de los requisitos o funcionalidades. Esta fracción se denomina como Sprint Backog, y debe cerrarse al principio de la iteración durante la reunión de planificación del Sprint. Scrum impone que ese conjunto de requisitos no puedan ser modificados durante la realización del Sprint.

5.7.3 Tablón de tareas

Habitualmente es posible utilizar una pizarra o un tablón de corcho para poner a la vista del equipo una lista de tareas pendientes, hechas o en proceso. Las tareas forman parte de los distintos requisitos (historias de usuario).

Figura 14.Ejemplo de Tablón de tareas.

5.7.4 Diagramas Burn-Down

La forma clásica de representar el grado de avance de los proyectos de desarrollo, como herencia recogida de la ingeniería civil, ha acabado siendo el popular diagrama de Gantt. Estos diagramas presentan diversos inconvenientes. En el contexto de Scrum no resultaría práctico representar las veinte o treinta tareas como barras horizontales, ya que daría lugar a un diagrama ilegible. La metodología Scrum propone una forma de representar el grado de progreso de un proyecto mediante los denominados diagramas Burn-Down. El eje vertical representa las tareas pendientes, y el eje horizontal el tiempo. Una línea discontinua refleja la progresión teórica o ideal del proyecto, en contraste con los datos reales en cada periodo, unidos por segmentos.

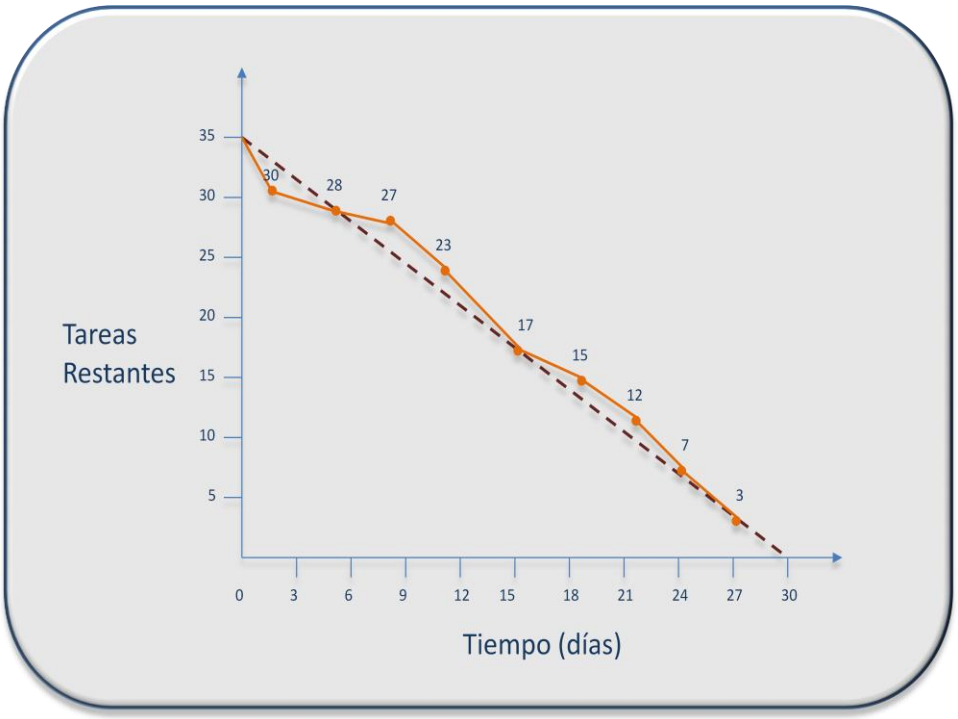


Figura 15.Ejemplo de diagrama Burn-Down.

Capítulo 6

Desarrollo de la aplicación AAS11

6.1 Introducción

En este capítulo realizará una exposición del proceso de desarrollo seguido para implementar la aplicación Automatic Audit System Oracle 11g (AAS11) que se utilizará como herramienta aplicable sobre entornos en los que se encuentre instalado el sgbdr Oracle 11g. La aplicación permitirá llevar a cabo una evaluación del estado de un determinado sistema, planteando una serie de cuestiones de auditoría y de tuning, que deben ser validadas por parte del usuario. Una vez respondidas la totalidad de cuestiones, se podrá generar un informe de auditoría que constituirá un valioso instrumento, a disposición del auditor, que puede utilizarse como referencia en la elaboración del informe de auditoría final.

Para llevar a cabo el desarrollo de la aplicación se utilizará la metodología de desarrollo ágil Scrum que implicará la consecución de los siguientes pasos:

- Identificación de requisitos software.
- Formalización de requisitos software en el Product Backlog.
- Realización de cada uno de los Sprints, que llevarán asociados:
    - Aplicación de técnicas de análisis.
    - Aplicación de técnicas de diseño.
    - Codificación de funcionalidades.
    - Prueba de cada una de las funcionalidades.

La sucesión de los distintos Sprints permitirá mostrar un proceso iterativo e incremental donde el sistema AAS11 alcanzará paulatinamente mayor número de funcionalidades hasta conseguir un producto acorde con la totalidad de requisitos establecidos.

6.2 Especificación de requisitos software

En los siguientes puntos se detallará la especificación de requisitos software que se ha tomado como base para implementar la aplicación AAS11. En el primer punto, se comentarán los fundamentos que se han empleado como referencia para la realización de la aplicación, mientras que en el segundo punto, se formalizará la especificación de requisitos elaborando el Product Backlog utilizado para dirigir la construcción del sistema.

6.2.1 Fundamentos de la especificación de requisitos

Con el objetivo de diseñar la especificación de requisitos de la manera más adecuada, se ha tomado como base una lista de comprobación aplicable a entornos basados en el sgbdr Oracle 11g procedente del CIS[76] (Center for Internet Security) denominada CIS Benchmark para Oracle [Cec11]. Esta lista de comprobación forma parte de un conjunto de herramientas de evaluación, datos y otros servicios, disponibles para la totalidad de usuarios. El conjunto de recomendaciones contenidas en esta lista de comprobación para el sgbdr de Oracle en su versión 11g, son el resultado de un proceso de construcción en el que han colaborado una gran cantidad de expertos en seguridad de Oracle. Esta lista de comprobación se divide en un número de secciones, y cada sección está constituida por una lista de requisitos o cuestiones que deben ser evaluadas. Cada cuestión o requisito incluye una descripción del requisito, la acción o recomendación asociada al establecimiento de valor de parámetros asociados al requisito, comentarios sobre la versión de Oracle sobre la que se puede aplicar y adicionalmente, si es aplicable sobre entornos Unix, Windows o ambos. Las secciones principales en las que se divide esta lista de comprobación se exponen a continuación:

1. Configuración específica de sistema operativo: en esta sección se mencionan requisitos asociados a la instalación de Oracle en los sistemas operativos Unix y Windows como por ejemplo, la verificación de permisos sobre determinados ficheros o la utilización de cuentas de acceso restringido.

2. Instalación y parches: a este nivel se establecen requisitos y cuestiones relativas a la consideración de restricciones durante el proceso de instalación, verificación de la utilización de la última versión de parches disponibles, inhabilitación y desinstalación de ciertas herramientas que podrían comprometer la seguridad…

3. Directorio de Oracle y permisos sobre ficheros: en este punto se plantean requisitos relativos al establecimiento de permisos sobre determinados tipos de ficheros y ajustes de ciertos parámetros.

[76] El objetivo del Centro de Seguridad para Internet (CIS) es mejorar la preparación y respuesta de seguridad de las entidades del sector público y privado, intentando alcanzar un compromiso de excelencia a través de la colaboración.

4. Parámetros de configuración de Oracle: aquí se establecen consideraciones relativas al establecimiento de valores en parámetros de los ficheros init.ora, listener.ora, sqlnet.ora y cman.ora. Valores adecuados sobre estos parámetros dotarán de mayor nivel de seguridad al sgbdr.

5. Configuración específica sobre cifrado: en este punto la mayoría de requisitos o cuestiones se relacionan con aspectos de cifrado utilizables a través de OAS[77].

6. Arranque y parada: en esta sección se comentan requisitos relativos a la gestión de colas en mensajes asíncronos y a la gestión de la caché.

7. Backup y recuperación ante desastres: aquí se comentan requisitos relativos a los ficheros de control, ficheros de actualización, copias de seguridad y cuestiones asociadas a la necesidad de la utilización de Oracle Failsafe como elemento que aumenta la seguridad.

8. Parametrización de puesta a punto del perfil de usuario de Oracle: en este punto se comentan gran cantidad de parámetros asociados al perfil de base de datos para usuarios Oracle.

9. Parametrización de acceso al perfil de usuario de Oracle: en el punto 9 se comentan requisitos que establecen restricciones con respecto al acceso a Tablespaces y con respecto al acceso a determinadas tablas, vistas, sinónimos y privilegios.

10. Enterprise Manager/Grid Control/Agentes: en este punto se detallan requisitos asociados al Enterprise Manager[78] y a Grid Control[79]. Adicionalmente, se comenta algún aspecto relativo al proceso de instalación del propio sgbdr.

11. Elementos relevantes para subsistemas específicos: en este punto se comentan ADDM[80], AMM[81] y AWR[82]. Por otra parte, se considera la posibilidad de utilizar la auditoría de grano fino como elemento susceptible de aplicarse sobre determinados objetos de la base de datos.

[77] Oracle Advanced Security (OAS) [Ora07] habilita características de seguridad facilitando el cumplimiento regulatorio al proteger los datos sensibles – en la red, en los medios de copia de seguridad o dentro de la base de datos – de la divulgación no autorizada.
[78] Oracle Enterprise Manager (OEM) es una herramienta que permite la monitorización de base de datos facilitando labores de administración.
[79] Oracle Enterprise Manager Grid Control es una herramienta que permite realizar labores de monitorización y administración en entornos distribuidos.
[80] The Automatic Database Diagnostic Monitor (ADDM) [Orabase1] es una herramienta que permite analizar los datos del Automatic Workload Repository (AWR) para identificar posibles cuellos de botella asociados con el rendimiento. Para cada problema identificado, se intenta determinar la causa base y se formulan recomendaciones para corregir el problema.
[81] Automatic Memory Management (AMM) [Burleson] es una característica que automáticamente reajusta el tamaño de las principales áreas de memoria utilizadas en el sgbdr de Oracle (db_cache_size, shared_pool_size, large_pool_size, java_pool_size) en base a la carga de trabajo existente.
[82] Automatic Workload Repository (AWR) [Cha08] se incorporó a partir de la versión del sgbdr Oracle 10g y se utiliza para recopilar, procesar y mantener estadísticas de la base de datos. Esta herramienta se basa en la utilización de capturas (snapshot) de determinados datos de la propia base de datos.

12. Políticas generales y procedimientos: en esta sección se comentan ciertos requerimientos de carácter general relacionados con la instalación de Oracle, revisión de los ficheros de alerta, comprobación de grupos de usuarios, localización de ficheros físicos en el soporte correspondiente…

13. Políticas de auditoría y procedimientos: en la sección 13 se detallan cuestiones de auditoría que deben considerarse en toda base de datos para permitir establecer un seguimiento adecuado de su actividad y evitar posibles problemas.

14. Apéndice A. Configuración adicional: en este punto aparecen requerimientos relacionados con la utilización Oracle Label Security y se realizan consideraciones adicionales con respecto a los soportes magnéticos, a los procedimientos de recuperación y a la revisión de routers y firewalls.

6.2.1.1 Relación establecida con el estándar ISO/IEC 27002

Tras realizar un resumen general de la lista de recomendación CIS Benchmark para Oracle [Cec11] se procederá a establecer una correspondencia entre dicha lista y los apartados asociados al punto 3.4.1 del Tema 3 "Estándar Internacional ISO/IEC 27002 " en el que se han descrito aquellos puntos que tienen particular relación con los sgbdr. En las siguientes tablas se muestran las relaciones identificadas:

Sección CIS Benchmark Oracle	Puntos ISO/IEC 27002	Comentarios
1.Configuración específica de sistema operativo	3.4.1.7 Gestión de los usuarios	A pesar de que en este punto no se hace referencia al sgbdr de Oracle sí que se establecen políticas para restringir el acceso a usuarios a nivel del sistema operativo.
2.Instalación y parches	3.4.1.6 Monitorización	Aunque en la monitorización se incide más en aspectos relacionados con la auditoría, destacamos en este punto que el sgbdr debe permanecer actualizado para que estos mecanismos funcionen correctamente.
3.Directorio de Oracle y permisos sobre ficheros	3.4.1.1 Clasificación de la información 3.4.1.7 Gestión de los usuarios	En esta sección se establecen limitaciones sobre el acceso a determinados ficheros y sobre la configuración de parámetros de Oracle, por lo que se relaciona con la clasificación de la información y con la gestión de usuarios para controlar el acceso a la información.

Tabla 7.Relación entre secciones 1, 2 y 3 de CIS Benchmark para Oracle e ISO/IEC 27002.

Sección CIS Benchmark Oracle	Puntos ISO/IEC 27002	Comentarios
4.Parámetros de configuración de Oracle	3.4.1.7 Gestión de los usuarios	Los valores de los parámetros que se establecen en los requisitos de esta sección están destinados a restringir las posibilidades de acceso al sgbdr de Oracle.
5.Configuración específica sobre cifrado	3.4.1.4 Respaldo o backup 3.4.1.5 Gestión de la seguridad de la red 3.4.1.8 Controles criptográficos 3.4.1.9 Cumplimiento de los requerimientos legales	Las herramientas y los algoritmos de cifrado utilizados en el sgbdr se pueden aplicar para aumentar la seguridad en las copias de seguridad realizadas, en el tráfico de red y sobre la propia información almacenada en la base de datos , con la finalidad de cumplir requisitos legales en materia de protección de datos.
6.Arranque y parada	3.4.1.6 Monitorización	En este punto se comentan ciertos aspectos susceptibles de monitorizarse como el vaciado de las cachés durante el cierre de la instancia.
7.Backup y recuperación ante desastres	3.4.1.4 Respaldo o backup	Los requisitos considerados en esta sección están fundamentalmente relacionados con la replicación de los ficheros de control, de Redo logs, copias de seguridad de los ficheros de Archivelog y utilización de Oracle Failsafe.
8.Parametrización de puesta a punto del perfil de usuario de Oracle	3.4.1.7 Gestión de acceso de los usuarios	En este punto se establece el valor de parámetros críticos en la configuración del perfil de usuario de Oracle, que determinarán y establecerán restricciones relativas a los recursos disponibles para cada usuario asociado a un determinado perfil.
9.Parametrización de acceso al perfil de usuario de Oracle	3.4.1.1 Clasificación de la información. 3.4.1.2 Condiciones previas a la contratación. 3.4.1.3 Terminación o cambio de empleo.	Sobre esta sección aparecen requisitos que limitan el acceso a tablas y vistas. Adicionalmente estos requisitos establecen restricciones sobre privilegios, roles y se limita el acceso sobre determinados paquetes de base de datos.

Tabla 8.Relación entre secciones 4, 5, 6, 7, 8 y 9 de CIS Benchmark para Oracle e ISO/IEC 27002.

Sección CIS Benchmark Oracle	Puntos ISO/IEC 27002	Comentarios
10.Enterprise Manager/Grid Control/Agentes	3.4.1.6 Monitorización	En esta sección aparecen requisitos asociados a varias herramientas de monitorización que pueden ser utilizadas con Oracle.
11.Elementos relevantes para subsistemas específicos	3.4.1.6 Monitorización	En esta sección aparecen requisitos asociados a ADDM, AWR y AMM que constituyen herramientas que permiten detectar y corregir problemas de eficiencia en la base de datos.
12.Políticas generales y procedimientos	3.4.1.6 Monitorización 3.4.1.7 Gestión de acceso de los usuarios 3.4.1.8 Controles criptográficos	En la sección de políticas generales y procedimientos aparecen varios requisitos relacionados con múltiples aspectos del sgbdr como son el procedimiento de instalación, aplicación de cifrado sobre información relevante de la base de datos, monitorización de los ficheros de logon, monitorización de los ficheros de log y establecimiento de restricciones sobre el acceso a determinados objetos a usuarios con privilegios.
13.Políticas de auditoría y procedimientos	3.4.1.6 Monitorización	Fundamentalmente en esta sección aparecen requerimientos asociados a la auditoría de determinados objetos de la base de datos sobre los que se pueden aplicar los distintos tipos de auditoría que aparecen definidos en el punto de Monitorización.
14.Apéndice A. Configuración adicional	3.4.1.4 Respaldo o backup. 3.4.1.5 Gestión de la seguridad de la red	En este punto se destacan requisitos asociados a la utilización de Oracle Label Security, requerimientos asociados a la verificación de la copia de seguridad y ciertos aspectos a considerar con respecto a la configuración de routers y firewalls.

Tabla 9.Relación entre secciones 10, 11, 12, 13 y 14 de CIS Benchmark para Oracle e ISO/IEC 27002.

6.2.2 Elaboración del Product Backlog

A continuación se expone la lista de requisitos de carácter general que se han establecido inicialmente. Esta lista aparece divida en tablas. En cada tabla se detallan cada uno de los requisitos que se han identificado. Cada uno de estos requisitos se define a través de los siguientes campos:

- ID: Identificador unívoco del requisito. Este identificador seguirá el siguiente patrón PB-*N*-*XXX* donde *N* identifica el Sprint en el que se ha establecido el requisito y *XXX* es un número de tres dígitos que identifica de forma unívoca el requisito.

- Nombre: nombre que identifica el requisito.

- Descripción: descripción asociada al requisito.

- Estabilidad: indica la posibilidad de que el requisito sea modificado por el Equipo de desarrollo a propuesta del Product Owner.

- Prioridad: prioridad establecida en el requisito, que determinará su incorporación en la aplicación AAS11. El rango de prioridades originalmente establecido es entre 0 y 1. Un valor mayor indica más prioridad.

- Coste: número utilizado para representar el coste asociado a la implementación del requisito. Se ha establecido un rango de coste entre 1 y 100. Un valor mayor indica más coste.

- Completado: porcentaje global del requisito que ha sido completado.

- Sprint 1: porcentaje del requisito completado en el Sprint 1.

- Sprint 2: porcentaje del requisito completado en el Sprint 2.

- Sprint 3: porcentaje del requisito completado en el Sprint 3.

- Sprint 4: porcentaje del requisito completado en el Sprint 4.

A continuación se exponen las tablas que contienen cada uno de los requisitos identificados:

PB-0-001							
Nombre:	Interfaz web						
Descripción:	La aplicación debe ser accesible para los usuarios utilizando únicamente un navegador. Dispondrá de una interfaz web que permitirá su utilización completa. La interfaz debe ser validada por los futuros usuarios de la aplicación.						
Estabilidad:	☐ Alta		☐ Media		☒ Baja		
Prioridad:	0,9		Coste:		70		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 10.Requisito PB-0-001.

PB-0-002							
Nombre:	Multiusuario						
Descripción:	La aplicación debe ser multiusuario, por lo que debe soportar la gestión completa de los mismos (altas, bajas y modificaciones).						
Estabilidad:	☒ Alta		☐ Media		☐ Baja		
Prioridad:	0,8		Coste:		25		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 11.Requisito PB-0-002.

PB-0-003							
Nombre:	Perfiles de usuario						
Descripción:	La aplicación dispondrá de varios perfiles. A priori "Administrador" y "Usuario". El perfil "Administrador" permitirá, además de las funcionalidades propias de la aplicación, llevar a cabo labores de parametrización y configuración.						
Estabilidad:	☐ Alta		☒ Media		☐ Baja		
Prioridad:	0,8		Coste:		18		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 12.Requisito PB-0-003.

PB-0-004							
Nombre:	Definición de cuestiones de auditoría						
Descripción:	La aplicación debe permitir la definición de cuestiones de auditoría. Una definición de una cuestión de auditoría constará de los siguientes campos: - Descripción de la cuestión. - Comentario asociado a la cuestión. - Respuesta afirmativa a la cuestión. - Respuesta negativa a la cuestión. - Sentencia de base de datos asociada a la cuestión. - Comentario asociado a la sentencia de base de datos. - Sistema operativo sobre el que es aplicable. - Sección a la que se asocia la cuestión.						
Estabilidad:	☐ Alta		■ Media		☐ Baja		
Prioridad:	0,8		Coste:		30		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 13.Requisito PB-0-004.

PB-0-005							
Nombre:	Gestión de cuestiones de auditoría						
Descripción:	La aplicación debe permitir la gestión de las cuestiones de auditoría de las que se deben componer los test, por lo que se habilitará la posibilidad de realizar las acciones de alta, baja y modificación sobre dichas cuestiones. Esta acción únicamente podrá ser realizada por usuarios con perfil de "Administrador".						
Estabilidad:	■ Alta		☐ Media		☐ Baja		
Prioridad:	0,7		Coste:		40		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 14.Requisito PB-0-005.

PB-0-006							
Nombre:	Clasificación de cuestiones de auditoría en secciones						
Descripción:	La aplicación permitirá llevar a cabo la clasificación de las cuestiones de auditoría en secciones para facilitar una asignación posterior a los usuarios. Esta acción únicamente podrá ser realizada por usuarios con perfil de "Administrador".						
Estabilidad:	■ Alta		☐ Media		☐ Baja		
Prioridad:	0,6		Coste:		10		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 15.Requisito PB-0-006.

PB-0-007							
Nombre:	Configuraciones particularizadas de cuestiones de auditoría						
Descripción:	La aplicación permitirá llevar a cabo configuraciones particularizadas para los usuarios, de tal forma que para cada uno de ellos se podrán construir cuestionarios con cuestiones de auditoría personalizadas. Esta acción podrá ser realizada por cualquier usuario, independientemente de su perfil.						
Estabilidad:	■ Alta		☐ Media		☐ Baja		
Prioridad:	0,5		Coste:		30		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 16.Requisito PB-0-007.

PB-0-008							
Nombre:	Definición de cuestiones de tuning						
Descripción:	La aplicación debe permitir la definición de cuestiones de tuning. Una definición de una cuestión de tuning constará de los siguientes campos: - Descripción de la cuestión de tuning. - Sentencia de base de datos asociada a la cuestión de tuning. - Comentario asociado a la sentencia de base de datos aplicable en la cuestión de tuning. - Gráfico utilizado para representar los datos de la sentencia.						
Estabilidad:	☐ Alta		☐ Media		■ Baja		
Prioridad:	0,8		Coste:		30		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 17.Requisito PB-0-008.

PB-0-009							
Nombre:	Gestión de cuestiones de tuning						
Descripción:	La aplicación debe permitir la gestión de las cuestiones de tuning que deben componer el test de tuning por lo que habilitará la posibilidad de realizar las acciones de alta, baja y modificación de dichas cuestiones. Esta acción únicamente podrá ser realizada por usuarios con perfil “Administrador”.						
Estabilidad:	■ Alta		☐ Media		☐ Baja		
Prioridad:	0,7		Coste:		40		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 18.Requisito PB-0-009.

PB-0-010							
Nombre:	Clasificación de las cuestiones de tuning en apartados						
Descripción:	La aplicación permitirá llevar a cabo la clasificación de las cuestiones de tuning en apartados para facilitar una asignación posterior de las mismas a los usuarios. Esta acción únicamente podrá ser realizada por usuarios con perfil "Administrador".						
Estabilidad:	■ Alta		☐ Media		☐ Baja		
Prioridad:	0,6		Coste:		10		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 19.Requisito PB-0-010.

PB-0-011							
Nombre:	Configuraciones particularizadas de cuestiones de tuning						
Descripción:	La aplicación permitirá llevar a cabo configuraciones particularizadas para los usuarios, de tal forma que para cada uno de ellos se podrán construir cuestionarios con cuestiones de tuning particularizadas. Esta acción podrá ser realizada por cualquier usuario, independientemente de su perfil.						
Estabilidad:	☐ Alta		☐ Media		■ Baja		
Prioridad:	0,5		Coste:		30		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 20.Requisito PB-0-011.

PB-0-012							
Nombre:	Elaboración de test						
Descripción:	La aplicación permitirá a la totalidad de usuarios (independientemente de su perfil) realizar los test con las cuestiones de auditoría y de tuning que tengan asignadas, con el objetivo de permitir la generación del informe de auditoría asociado a la combinación de estos cuestionarios.						
Estabilidad:	☐ Alta		■ Media		☐ Baja		
Prioridad:	0,8		Coste:		60		
Completado:	0%						
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 21.Requisito PB-0-012.

PB-0-013							
Nombre:		Gestión de test					
Descripción:		La aplicación habilitará la posibilidad a cada uno de los usuarios de gestionar cada uno de sus test permitiendo tanto su creación como su almacenamiento.					
Estabilidad:		☐ Alta		■ Media		☐ Baja	
Prioridad:		0,5		Coste:		40	
Completado:		0%					
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 22.Requisito PB-0-013.

PB-0-014							
Nombre:		Informes en formato Word					
Descripción:		La aplicación permitirá generar los informes de auditoría que se obtendrán como resultado de la realización de los Test en formato Word.					
Estabilidad:		■ Alta		☐ Media		☐ Baja	
Prioridad:		0,4		Coste:		50	
Completado:		0%					
Sprint 1:	0%	Sprint 2:	0%	Sprint 3:	0%	Sprint 4:	0%

Tabla 23.Requisito PB-0-014.

6.3 Sprint 1

6.3.1 Reunión de planificación del Sprint 1

Durante la reunión de planificación del Sprint 1 se escogen los siguientes requisitos para su implementación.

- PB-0-001 Interfaz web: la aplicación debe ser accesible para los usuarios utilizando únicamente un navegador. Dispondrá de una interfaz web que permitirá su utilización completa. La interfaz debe ser validada por los futuros usuarios de la aplicación.

- PB-0-002 Multiusuario: la aplicación debe ser multiusuario por lo que debe soportar la gestión completa de los mismos (altas, bajas y modificaciones).

- PB-0-003 Perfiles de usuario: la aplicación dispondrá de varios perfiles. A priori "Administrador" y "Usuario". El perfil "Administrador" permitirá, además de las acciones propias de la aplicación, llevar a cabo labores de parametrización y configuración.

El Sprint Backlog asociado al Sprint 1 se muestra a continuación:

ID	Nombre	Prioridad	Coste	Completado	Sprint 1
PB-0-001	Interfaz web	0.9	70	0	0
PB-0-002	Multiusuario	0.9	25	0	0
PB-0-003	Perfiles de usuario	0.8	18	0	0

Tabla 24.Sprint Backlog 1.

Sobre cada uno de estos requisitos se realizan las siguientes observaciones:

- PB-0-001: este requisito establece la necesidad de que la aplicación sea utilizable por cualquier usuario disponiendo únicamente de un navegador Web. A priori, se considera que es un requisito de baja estabilidad puesto que la propia interfaz debe ser validada por los usuarios finales de la aplicación. Tras plantear este requisito se determina que debe utilizarse una arquitectura cliente-servidor en la que múltiples usuarios podrán utilizar de forma simultánea el sistema y que adicionalmente, permitirá una gestión centralizada de la información. Se trata de un requisito transversal puesto que afecta al resto de componentes de la aplicación.

- PB-0-002: la aplicación deberá permitir la conexión simultánea de varios usuarios sobre una base de datos. Puesto que la aplicación podrá ser utilizada por múltiples usuarios, debe de existir un conjunto de opciones que permitan la gestión de los mismos. Estas opciones únicamente serán accesibles a usuarios con determinado perfil.

- PB-0-003: estrechamente asociado al requisito PB-0-002 aparece este requisito que permite la utilización de perfiles en la aplicación. Estos perfiles serán susceptibles de ser asignados a los usuarios y determinarán el acceso a las distintas opciones de las que se compone la misma.

Para satisfacer estos requisitos se identifican las tareas que deben llevarse a cabo:

ID	Nombre	Tareas asociadas
PB-0-001	Interfaz web	1. Selección de la arquitectura (framework, servidor de aplicaciones, herramientas de control y software subyacente). 2. Configuración y parametrización de aplicación. 3. Diseño de la capa de presentación de la aplicación. 4. Elaboración de un prototipo para realizar su validación.
PB-0-002	Multiusuario	1. Planteamiento del requisito utilizando un diagrama de casos de uso. 2. Generación de esquema de base de datos. 3. Planteamiento inicial del modelo de datos necesario para llevar a cabo la gestión de usuarios. 4. Implementación de la parte del modelo de datos que permite dotar de esta funcionalidad. 5. Carga de datos en base de datos. 6. Construcción de procedimiento de Login para permitir la entrada de usuarios en la aplicación. 7. Prueba de procedimiento de Login. 8. Construcción de opción de “Alta de Usuarios” en la aplicación. 9. Prueba de opción de “Alta de Usuarios”. 10. Construcción de opción de “Modificación de Usuarios” en la aplicación. 11. Prueba de opción de “Modificación de Usuarios”. 12. Construcción de opción de “Baja de Usuarios” en la aplicación. 13. Prueba de opción de “Baja de Usuarios”.
PB-0-003	Perfiles de usuario	1. Planteamiento del requisito utilizando un diagrama de casos de uso. 2. Introducción en el modelo de datos de la gestión de perfiles de usuario. 3. Carga de datos en la base de datos. 4. Construcción de la opción “Consulta de perfiles/menús/opciones”. 5. Prueba de la opción de consulta de “Consulta de perfiles/menús/opciones”.

Tabla 25.Relación entre requisitos y tareas identificadas en Sprint 1.

6.3.2 Tareas asociadas al requisito PB-0-001

6.3.2.1 Arquitectura de la aplicación

En el desarrollo del sistema se ha aplicado el patrón MVC (Model View Controller). Este patrón [Pav08] establece una separación entre los datos de una aplicación, la interfaz de usuario y la lógica de control en componentes distintos, de forma que las modificaciones realizadas sobre cualquier parte del sistema puedan ser acometidas con un mínimo impacto sobre el resto de componentes del sistema. Este patrón cumple perfectamente con la finalidad de modularizar un sistema. Los principales componentes del patrón MVC son:

- Modelo: constituye la representación específica de los datos con los que opera el sistema.

- Vista: habitualmente constituye la interfaz de usuario. Es la responsable de transformar el modelo en un elemento perceptible para el usuario, es decir, lleva a cabo una transformación de los datos para que el usuario pueda interpretarlos y constituyan información útil.

- Controlador: responsable de controlar el flujo y estado de la entrada de datos por parte del usuario.

En la siguiente figura se muestran los componentes del patrón MVC y su interrelación con las tecnologías y herramientas utilizadas en el desarrollo de la aplicación AAS11:

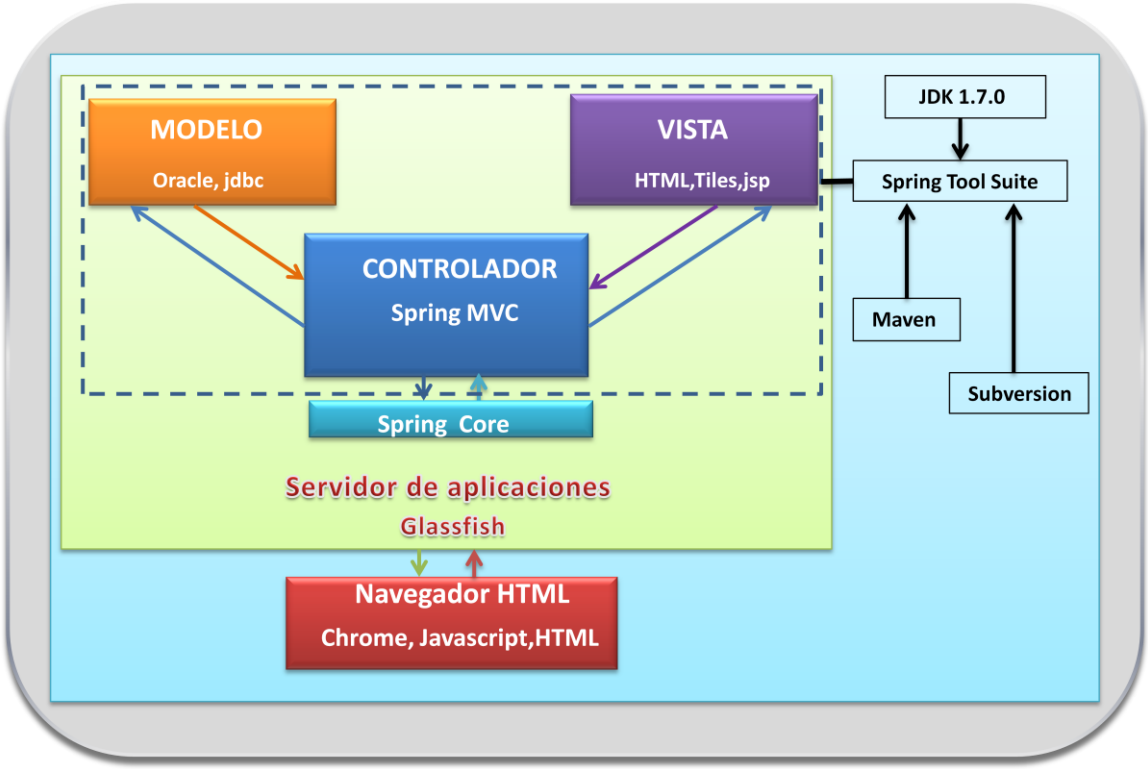


Figura 16.Modelo MVC y tecnologías asociadas.

La arquitectura de software se expone en los siguientes puntos:

- Sistema operativo: se ha seleccionado el sistema operativo de Microsoft Windows 7 debido a su gran difusión y aceptación por parte de los usuarios. Esta plataforma se utilizará como base para la instalación del servidor de aplicaciones y para la ejecución de los clientes asociados. *Versión: Windows 7 Ultimate*.

- Sgbdr Oracle: el sgbdr instalado es Oracle 11g. En el propio sgbdr se creará un esquema utilizado por la aplicación AAS11. Sobre el sgbdr de Oracle se plantearán tanto las cuestiones de auditoría como las cuestiones de tuning. *Versión: Oracle 11g Release 2 Enterprise Edition.*

- Para la implementación de la aplicación se utilizará el lenguaje JAVA, debido a sus características intrínsecas como la portabilidad, sencillez, robustez y seguridad. *Versión: JDK 1.7.0.*

- Como servidor de aplicaciones se empleará Glassfish. Este servidor de aplicaciones se utilizará como servidor Web y como contenedor de los componentes ejecutables vía Web (Servlets). *Versión: Glassfish Server Open Source Edition 3.0.1 (build 22).*

- El framework utilizado como base para la construcción de la aplicación será Spring. Se ha escogido este framework debido al gran número de extensiones y características que proporciona para construir aplicaciones web. *Versión: Spring 3.1.2.*

- Como entorno integrado de desarrollo se ha utilizado Spring Tool Suite puesto que está particularmente adaptado para el uso del framework Spring y proporciona a los desarrolladores todas las herramientas necesarias para la creación de aplicaciones Web. *Versión: Spring tool suite 3.1.0 RELEASE*.

- La herramienta software utilizada para la gestión y construcción del proyecto ha sido Maven. Esta herramienta se utiliza para describir el proyecto software a construir, estableciendo las dependencias entre módulos, componentes externos y el orden de construcción de los elementos. *Versión: Apache Maven 3.0.4.*

- El sistema de control de versiones seleccionado para el proyecto es Subversión. Un sistema de control de versiones facilita en gran medida el desarrollo colaborativo, distribuido y permite la recuperación de versiones anteriores cuando así se requiera. *Versión: Subversion 1.6.*

6.3.2.2 Configuración y parametrización de la aplicación

Como paso previo al desarrollo de la aplicación AAS11 se procede a establecer la configuración del servidor de aplicaciones Glassfish. Tras establecer esta configuración, se lleva a cabo la incorporación de todos los elementos asociados al desarrollo (packages, librerías…) utilizando Maven. Después de llevar a cabo esta acción, se determina la estructura de directorios que se utilizará para albergar el código fuente de la aplicación. Finalmente, se comienza a definir el contenido de los ficheros XML encargados de soportar la configuración de la aplicación.

6.3.2.2.1 Configuración del servidor de aplicaciones Glassfish

En las siguientes capturas de pantalla se expone la configuración relativa al conjunto de conexiones JDBC utilizado para el acceso directo a la base de datos. La configuración establecida está dividida en tres pestañas que contemplan propiedades del tipo "General", "Avanzado" y "Propiedades adicionales".

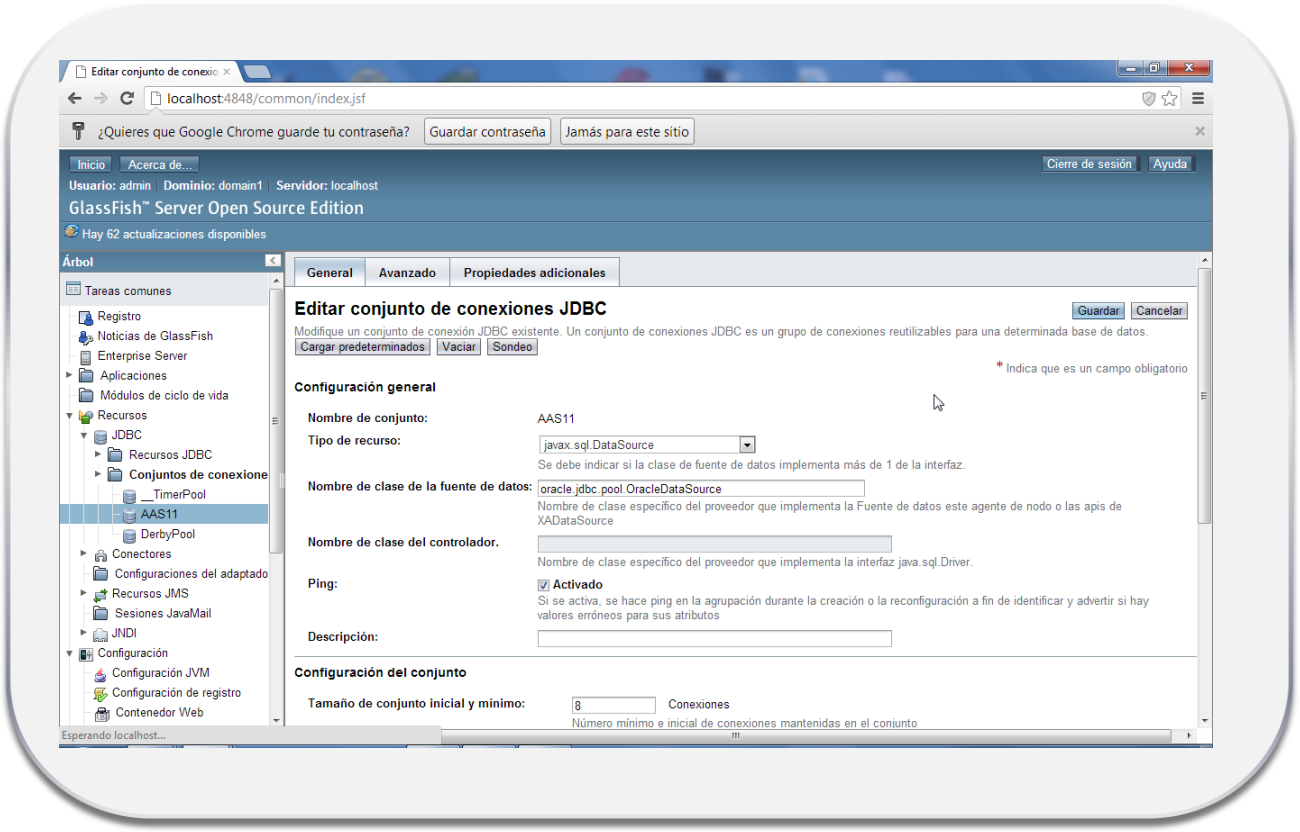


Figura 17. Conjunto de conexiones JDBC AAS11.Pestaña General.

En la pantalla que se muestra a continuación puede observarse como se han establecido valores para el "Nombre del conjunto", "Tipo de recurso" y para el "Nombre de la clase de la fuente de datos".

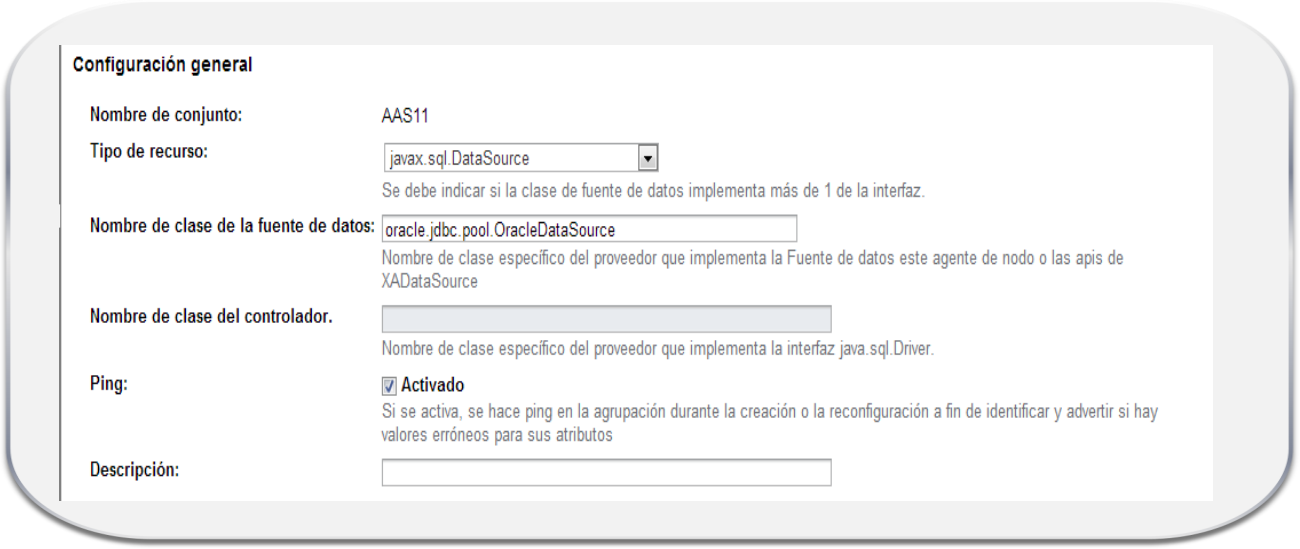


Figura 18. Detalle de la configuración general en el conjunto de conexiones AAS11.

Los valores que se muestran en la siguiente pantalla no han sido modificados con respecto a la configuración por defecto que se establece en el propio servidor.

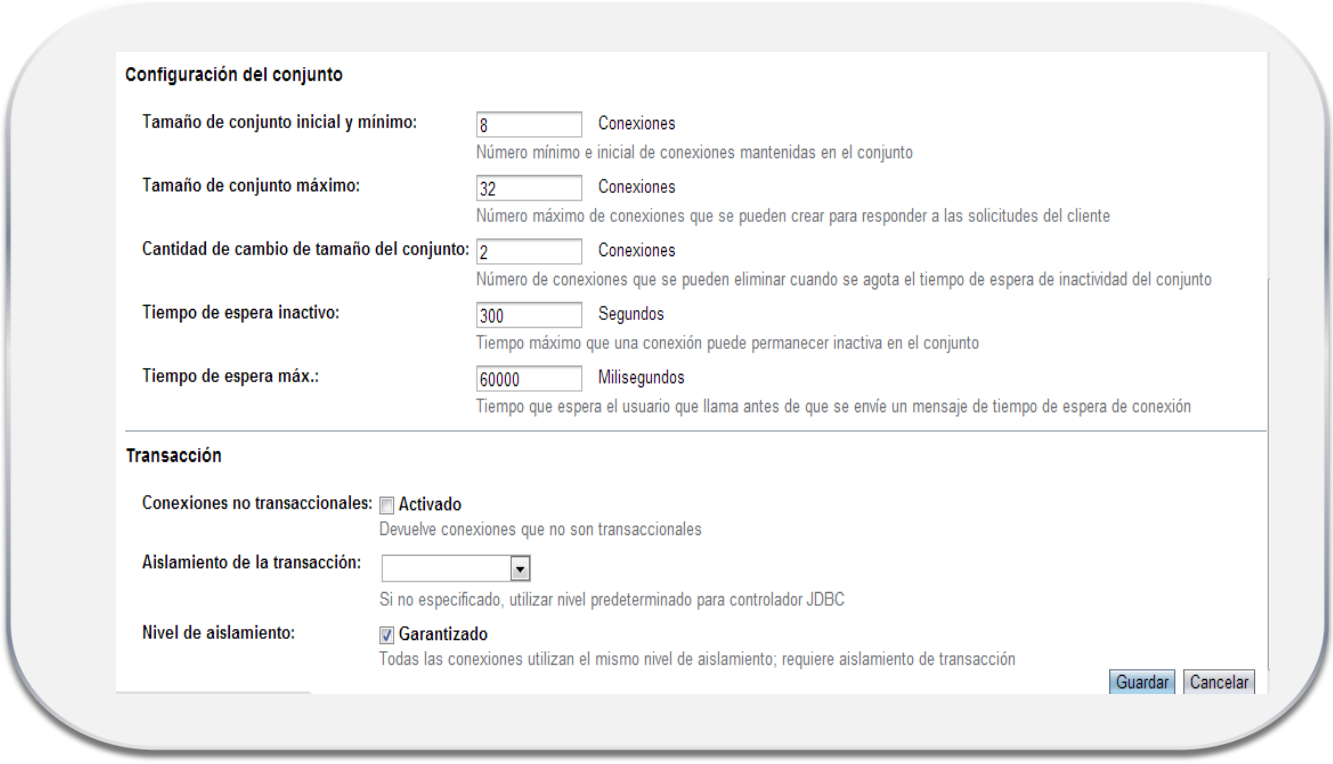


Figura 19. Detalle de la configuración del conjunto y de transacciones en el conjunto de conexiones AAS11.

En la siguiente pantalla aparecen los atributos avanzados del conjunto de conexiones JDBC. Sobre estos valores tampoco se ha realizado ninguna modificación con respecto a la configuración por defecto.

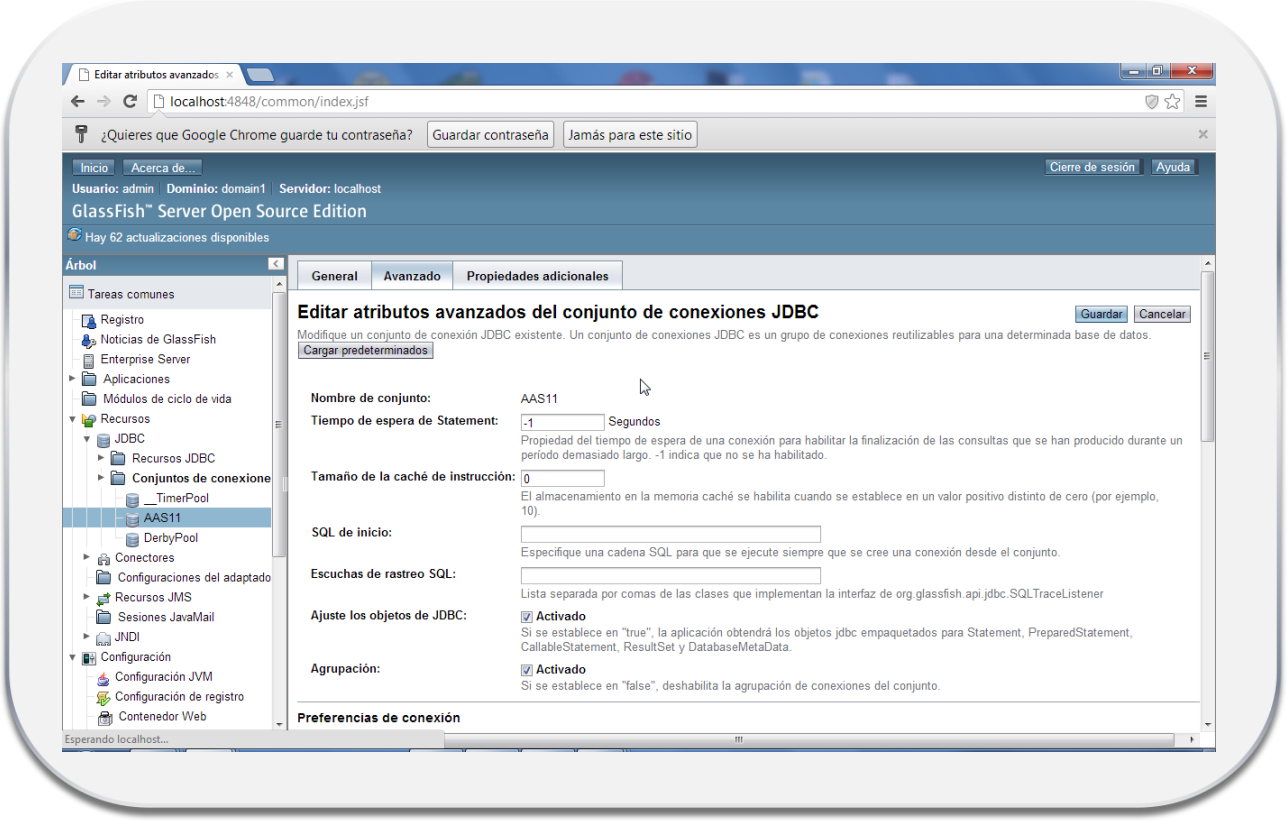


Figura 20. Conjunto de conexiones JDBC creado AAS11.Pestaña Avanzado.

En la pantalla que aparece a continuación se visualiza el detalle de los valores por defecto que asumen estos parámetros:

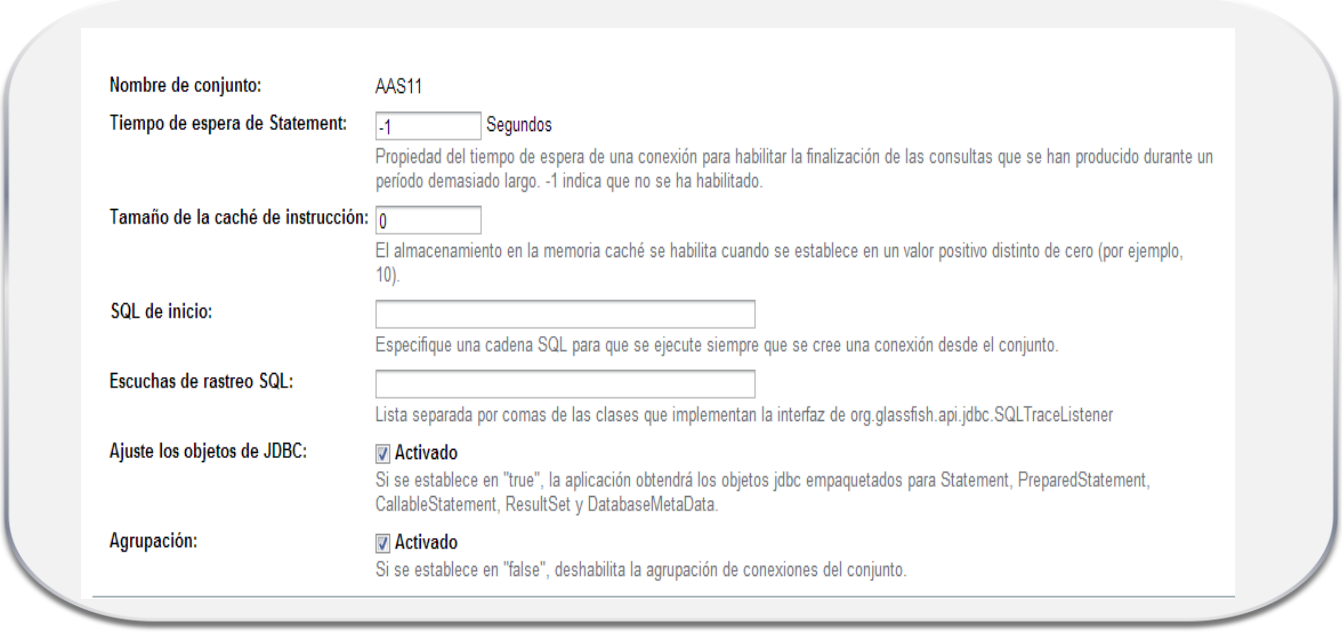


Figura 21. Detalle de la configuración avanzada en el conjunto de conexiones AAS11.

En las siguientes pantallas se visualiza el detalle de los valores por defecto que asumen los parámetros de configuración asociados a las "Preferencias de conexión" y la "Validación de la conexión":

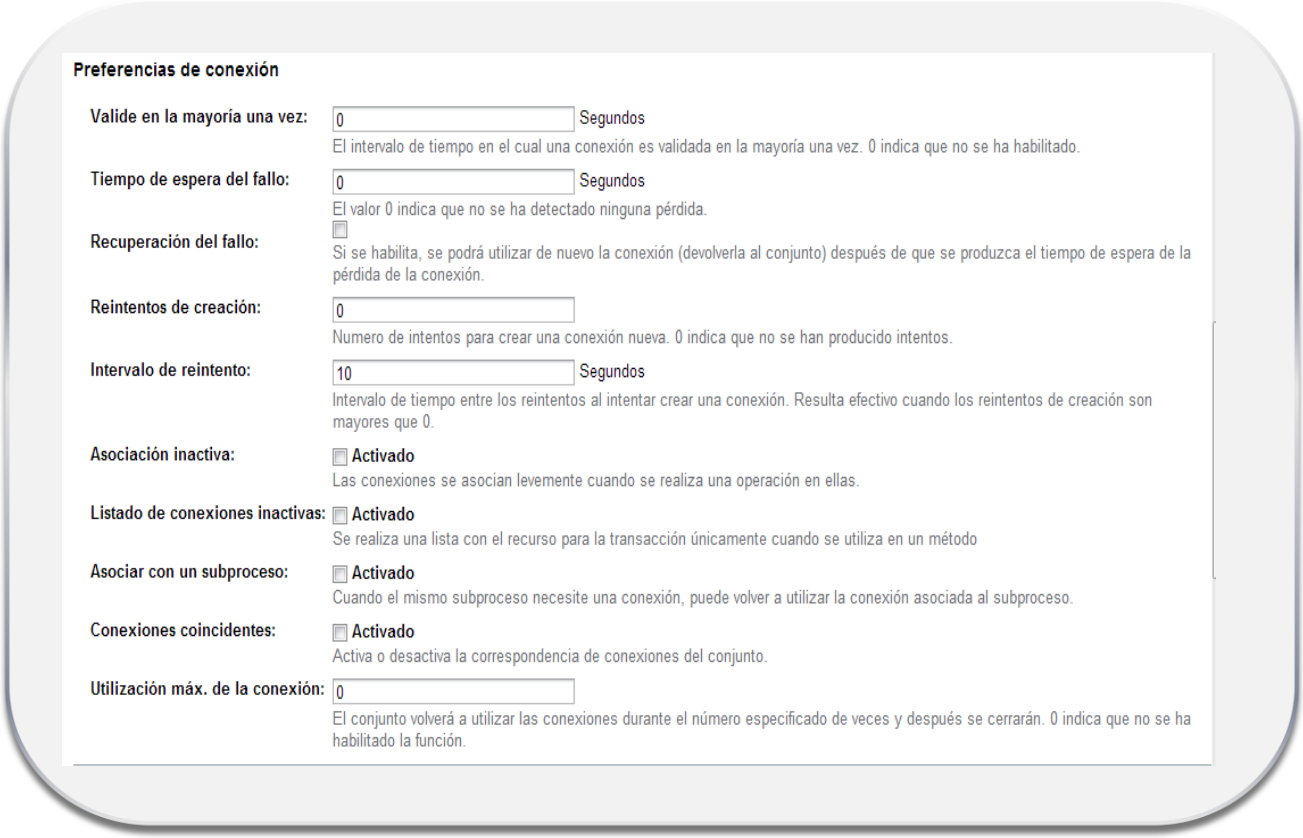


Figura 22. Detalle de preferencias de conexión en configuración avanzada del conjunto de conexiones AAS11.

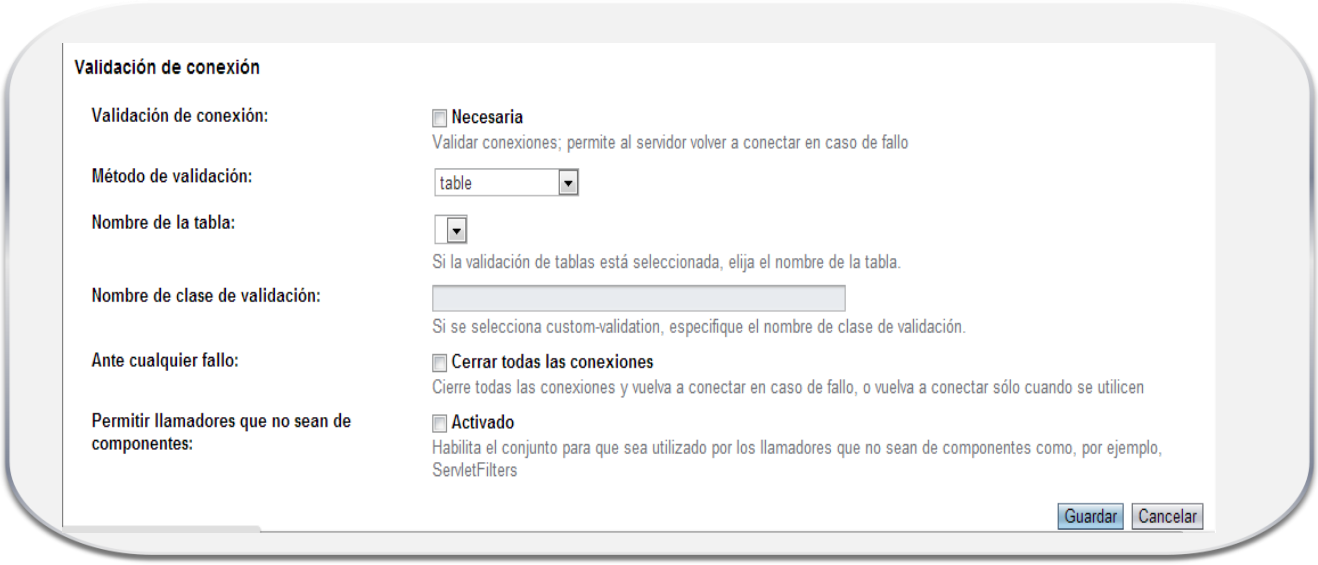


Figura 23. Detalle de validación de conexión en configuración avanzada del conjunto de conexiones AAS11.

En las pantallas que se muestran a continuación, aparecen el conjunto de parámetros asociados a la conexión al esquema de base de datos AAS11 que será el que utilice la aplicación. Los parámetros que aparecen detallados son la "URL" de acceso, el usuario "USER" y la palabra clave "PASSWORD":

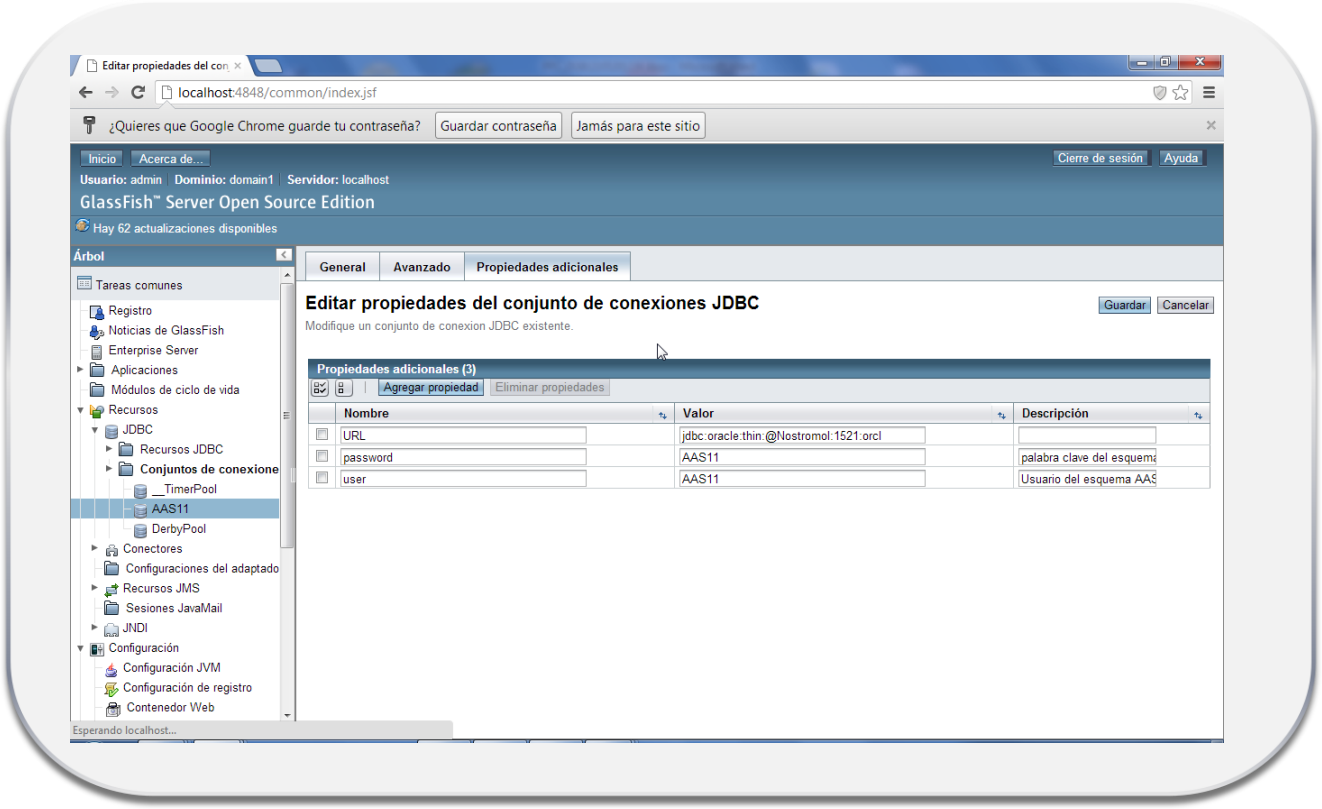


Figura 24. Conjunto de conexiones JDBC AAS11.Pestaña Propiedades adicionales.

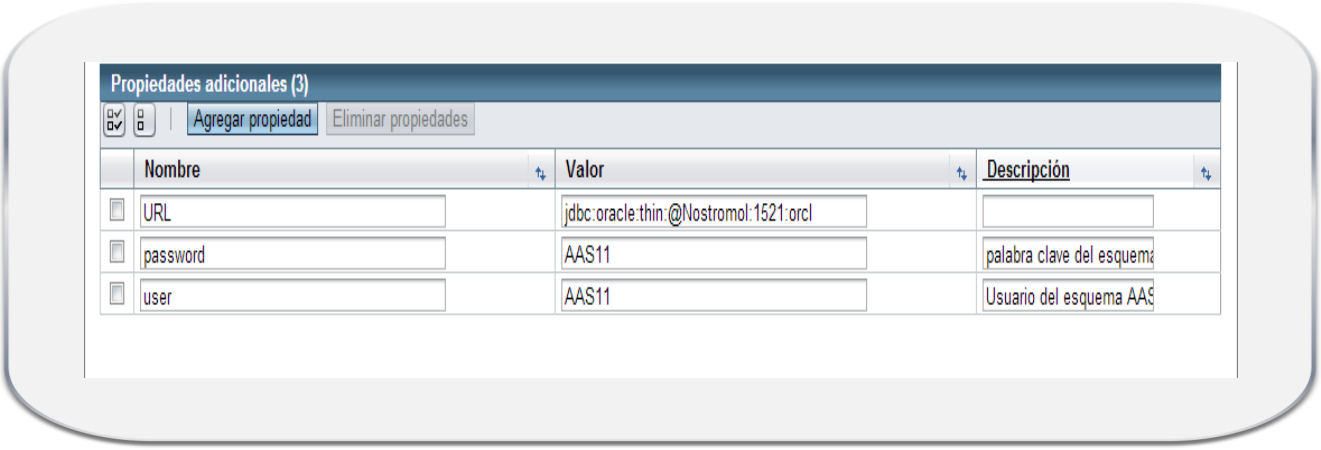


Figura 25. Detalle de las propiedades adicionales del conjunto de conexiones AAS11.

En las siguientes capturas de pantalla se expone la configuración relativa al recurso JDBC utilizable por la aplicación AAS11 para permitir un acceso directo al origen de datos:

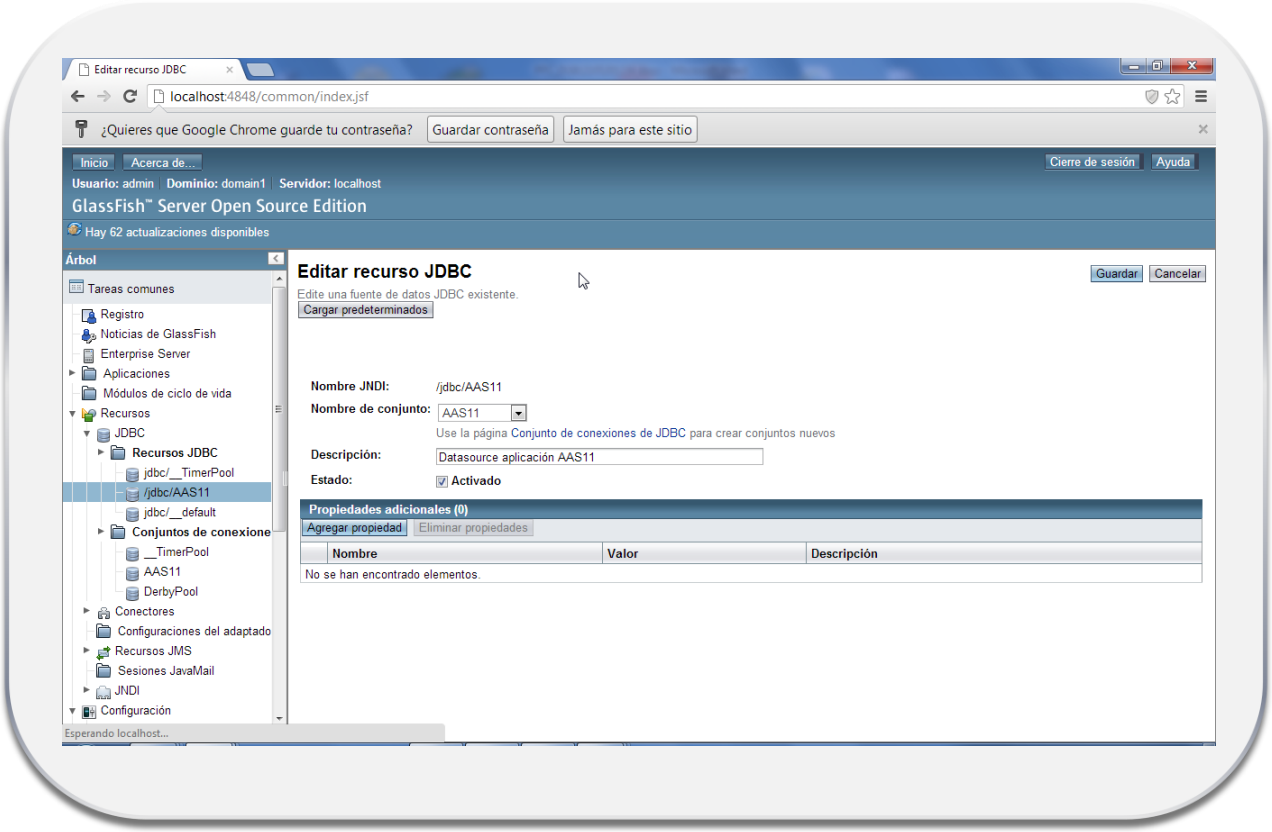


Figura 26. Recurso JDBC AAS11.

Figura 27. Detalle de propiedades JDBC AAS11.

6.3.2.2.2 Organización de ficheros fuente en directorios

En la siguiente figura aparece detallada la estructura de directorios utilizada en el desarrollo de la aplicación AAS11.

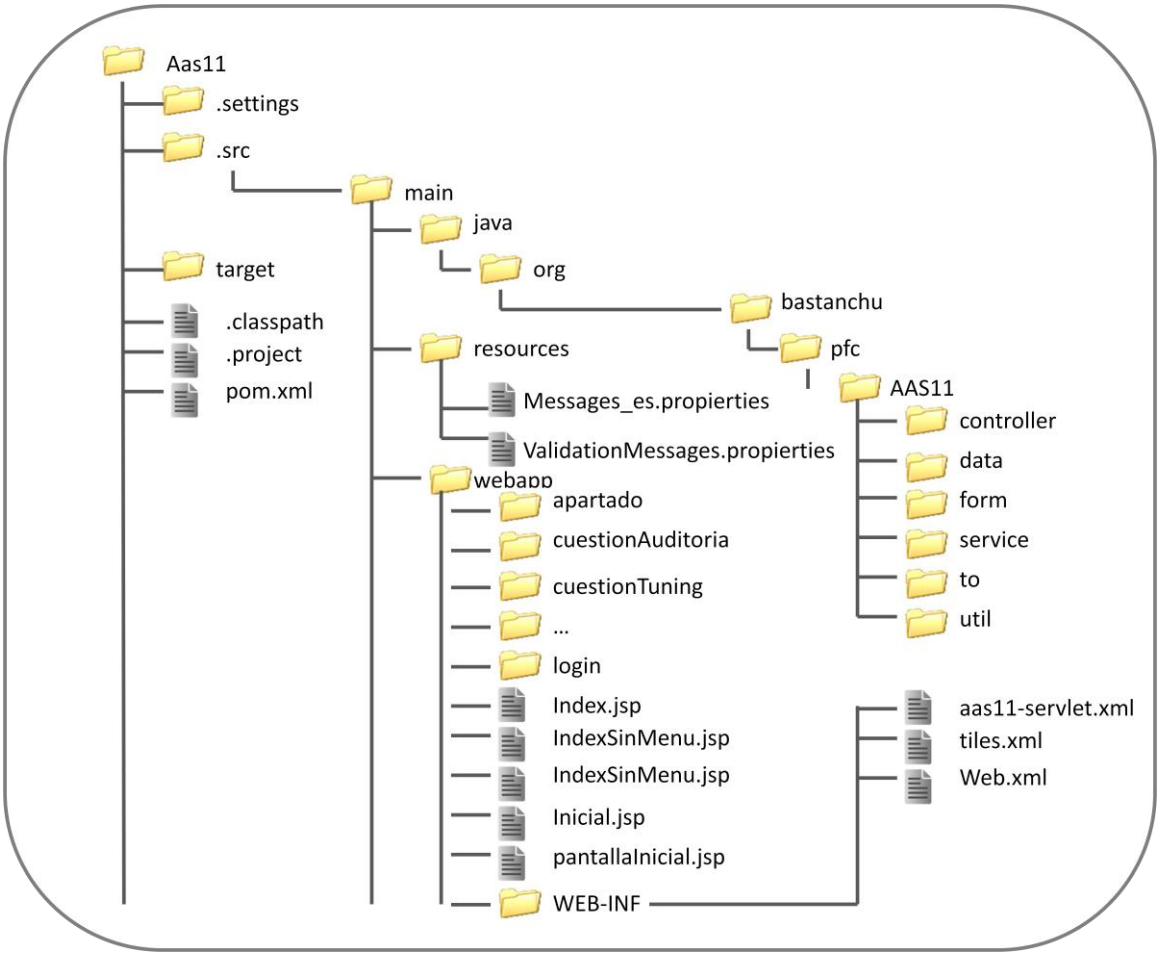


Figura 28. Estructura de directorios de la aplicación AAS11.

El contenido de cada uno de los directorios es el siguiente:

- Settings: directorio en el que el IDE utilizado (Spring Tool Suite en este caso) registra meta-información asociada al proyecto.

- Src: directorio en el que se almacenan los ficheros fuentes asociados al proyecto. Dentro de la jerarquía que se construye a partir de este directorio, aparece el directorio main que a su vez alberga una serie de subdirectorios principales:

    - Java: directorio en el que se organizan los ficheros fuentes java. Dentro de este directorio se construye una jerarquía que describe la dependencia de la aplicación dentro de la organización en la que se ha desarrollado. De esta forma aparece el directorio org (organización sin ánimo de lucro), bastanchu (subdivisión de la organización encargada de la elaboración del proyecto), pfc (categoría a la que pertenece el proyecto desarrollado) y por

último AAS11 (denominación de la aplicación desarrollada). A partir de este directorio se localizan los subdirectorios en los que se organizan los ficheros fuentes java, que se describen a continuación:

- Controller: directorio en el que se alojan las clases controller del proyecto, cuya labor es recoger las solicitudes realizadas por los usuarios (request) e interactuar con las clases pertenecientes a la vista y el modelo para resolver la solicitud realizada.
- Data: directorio que alberga la definición de interfaces y clases que permiten interactuar directamente con la base de datos y realizar las acciones de consulta y actualización sobre la propia base de datos. Estas clases, dentro del paradigma modelo-vista-controlador, pertenecerían al modelo.
- Form: clases utilizadas para la definición de datos intercambiados en los formularios utilizados por la aplicación AAS11.
- Service: directorio en el que se alojan las interfaces y clases utilizadas para definir e implementar los elementos relativos a cada funcionalidad. Este conjunto de clases e interfaces definen los servicios proporcionados internamente por la aplicación.
- TO: en este directorio se almacenan las clases utilizadas para la definición de los objetos de transferencia (Transfer Objects). Cada una de estas clases se compone de una serie de atributos y de los métodos getters y setters[83] asociados a cada uno de estos atributos. Representan a los datos en las maniobras de transferencia entre los controladores y los DAO´s (Data Access Object).
- Util: directorio en el que se almacenan clases genéricas utilizadas dentro del ámbito de la aplicación no vinculadas explícitamente a ningún elemento de arquitectura del sistema. Ejemplos de este tipo de clases podrían ser la clase utilizada para controlar la paginación o la clase utilizada para controlar la validación de usuarios.

- Resources: directorio en el que se organizan los ficheros de claves utilizados por la aplicación AAS11. En estos ficheros aparecen referenciados el conjunto de literales utilizados por la aplicación:
  - Messages_es.properties: definición de literales utilizados en las pantallas de la aplicación AAS11.
  - ValidationMessages.propierties: definición de literales utilizados a través de Hibernate Validator[84] para visualizar errores detectados en la entrada de valores en los formularios.

[83] Los métodos getters y setters se vinculan a propiedades visibles desde el exterior.
[84] Hibernate Validator [HIB] constituye una implementación del estándar JSR 303 para la validación de Beans. A través de este estándar es posible establecer validaciones en los campos de las clases utilizando anotaciones. Además permite la posibilidad de reemplazar estas validaciones mediante un descriptor XML.

```
#login
login.usuario=Usuario:
login.password=Password:
login.boton1=Entrar
login.mensajeDeCarga=Cargando

#indice principal
index.cuadroIdUsuario.usuario=Usuario:
index.cuadroIdUsuario.cadConexion=Conexi&oacute;n:
index.cuadroIdUsuario.perfil=Perfil:

#Botones del indice principal
index.boton1=Ayuda
index.boton2=Salir Sesi&oacute;n
...
```

Figura 29.Inicio del fichero Messages_es.propierties.

```
#Error login
error.login=Login fallido
#Usuarios
NotEmpty.usuarioForm.idUsuario=El identificador de usuario no puede estar vacío.
NotEmpty.usuarioForm.desUsuario=Debe introducir una descripción de usuario.
NotEmpty.usuarioForm.password=La clave no puede estar vacía.
NotEmpty.usuarioForm.confirmaPassword=La repetición de la clave no puede estar
vacía.
ValidPassword.usuarioForm.password=Debe coincidir la clave de acceso y la repetición
de la clave.
#Secciones
NotNullcionForm.idSeccion=El identificador de sección no puede estar vacío.
NotEmpty.seccionForm.tituloSeccion=El título de la sección no puede estar vacío.
NotEmpty.seccionForm.desSeccion=La descripción de la sección no puede estar vacía.
...
```

Figura 30.Inicio del fichero ValidationMessages.propierties.

- Webapp: directorio en el que se organizan los subdirectorios que albergan los JSP's de la aplicación. Este tipo de tecnología permite la generación de páginas web de forma dinámica. Los subdirectorios que aparecen en webapp aparecen organizados por funcionalidad. Adicionalmente a estos subdirectorios, aparecen ficheros utilizados de forma transversal en la aplicación AAS11:
  - Index.jsp: fichero utilizado para representar la cabecera de la aplicación AAS11 que incluye el menú principal de la aplicación.
  - IndexSinMenu.jsp: fichero que representa la cabecera de la aplicación AAS11 utilizado en las pantallas de visualización de mensajes, tras realizar una acción de actualización en la aplicación.
  - Inicial.jsp y PantallaInicial.jsp: ficheros que representan la pantalla inicial de la aplicación AAS11.

  Además de los directorios y ficheros enunciados, el directorio webapp contiene el subdirectorio WEB-INF que almacena los ficheros de configuración de la aplicación[85] AAS11.

- Target: en este directorio se generará el fichero WAR utilizado para desplegar la aplicación en el servidor de aplicaciones.

- .classpath: fichero que contiene la relación de librerías accesibles pare el editor de código y en general, para maniobras gestionadas dentro de este entorno, como la realización de compilaciones o despliegues.

- .project: fichero que contiene información relativa a la configuración general del proyecto dentro del entorno de desarrollo. Ejemplos de este tipo de información pueden ser versión de JDK o la configuración de la ubicación del repositorio remoto de gestión de versiones, en el caso de que se esté utilizando.

- Pom.xml: fichero de configuración XML asociado a la utilización de Maven.

[85] En el siguiente punto "Ficheros de configuración XML" se detallará el contenido de cada uno de los ficheros de configuración almacenados en el directorio WEB-INF.

6.3.2.2.3 Ficheros de configuración XML

Los ficheros de configuración XML se utilizarán para parametrizar y configurar determinados aspectos de la aplicación. A continuación se detallan cada uno de los ficheros de configuración XML utilizados en el desarrollo de la aplicación:

- POM.xml [AMP]: el Project Object Model o POM es la unidad fundamental de trabajo en Maven. Este es un fichero XML que contiene información sobre el proyecto y detalles de configuración utilizados por Maven para construir el proyecto. Por defecto contiene posibles configuraciones utilizables por una gran cantidad de proyectos. A continuación se expone un extracto del fichero de configuración utilizado en la aplicación AAS11:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
<modelVersion>4.0.0</modelVersion>
  <groupId>org.bastanchu.pfc</groupId>
  <artifactId>aas11</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>aas11 Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
                  <groupId>org.springframework</groupId>
                  <artifactId>spring-web</artifactId>
                  <version>3.0.5.RELEASE</version>
     </dependency>
     <build>
    <finalName>aas11</finalName>
    <plugins>
          <plugin>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>2.3.2</version>
             <configuration>
                  <source>1.6</source>
                  <target>1.6</target>
             </configuration>
          </plugin>
    </plugins>
  </build>
</project>
```

Figura 31.Extracto de fichero POM.xml.

- AAS11-servlet.xml [Wal11]: Spring es un marco de trabajo basado en contenedor. La configuración que contiene este fichero indica a Spring el contenido de dicho contenedor (beans[86]) y la forma de interconexión entre estos beans con el fin de que puedan trabajar de forma conjunta. A continuación se muestra un extracto del contenido del fichero AAS11-servlet.xml utilizado por la aplicación:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:mvc="http://www.springframework.org/schema/mvc"
        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:jee="http://www.springframework.org/schema/jee"
        xmlns:security="http://www.springframework.org/schema/security"
        xsi:schemaLocation="http://www.springframework.org/schema/mvc
                http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd
                http://www.springframework.org/schema/context
                http://www.springframework.org/schema/context/spring-context-3.0.xsd
                http://www.springframework.org/schema/beans
                http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
                http://www.springframework.org/schema/jee
                http://www.springframework.org/schema/jee/spring-jee-3.0.xsd
                http://www.springframework.org/schema/security
            http://www.springframework.org/schema/security/spring-security-3.0.3.xsd">
        <context:component-scan base-package="org.bastanchu.pfc.aas11" />
        <mvc:resources mapping="/resources/**"
            location="/resources/" />
    <mvc:annotation-driven/>
    <!-- Configuracion de tiles -->
    <bean id="viewResolver"
          class="org.springframework.web.servlet.view.UrlBasedViewResolver">
        <property name="viewClass">
        <value>org.springframework.web.servlet.view.tiles2.TilesView</value>
        </property>
    </bean>
    <bean id="tilesConfigurer"
          class="org.springframework.web.servlet.view.tiles2.TilesConfigurer">
         <property name="definitions">
        <list>
        <value>/WEB-INF/tiles.xml</value>
        </list>
         </property>
    </bean>

     . . .
</beans>
```

Figura 32.Extracto de fichero AAS11-servlet.xml.

[86] Un Bean [Wal11] es un componente software reutilizable. En diciembre del año 1996, Sun Microsystems publicó la especificación JavaBeans 1.00-A. Está especificación definía un modelo de componentes software para Java. En marzo de 1998 Sun publicó la versión 1.0 de la especificación Enterprise Java Beans.

- Tiles.xml [tiles]: Apache Tiles es un framework de plantillas utilizado para simplificar el desarrollo de interfaces de usuario en las aplicaciones web. Tiles permite la definición de fragmentos de página que pueden ser ensamblados en una página final en tiempo de ejecución. Estos fragmentos, denominados tiles, pueden ser utilizados para evitar repetir multitud de elementos dentro de las páginas, así como para desarrollar plantillas reutilizables que permitan mantener el aspecto de la aplicación. El fichero de configuración tiles.xml contendrá tanto la declaración de cada uno de los tiles o fragmentos utilizados por la aplicación como sus dependencias. A continuación podemos ver un fragmento del fichero de configuración tiles.xml utilizado por la aplicación AAS11:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE tiles-definitions PUBLIC
      "-//Apache Software Foundation//DTD Tiles Configuration 2.0//EN"
      "http://tiles.apache.org/dtds/tiles-config_2_0.dtd">
<tiles-definitions>
          <definition name="plantillaBasica"
                    template="/index.jsp">
                     <put-attribute name="titulo" value=""/>
                     <put-attribute name="cuerpo" value=""/>
          </definition>
                     <definition name="plantillaSinMenu"
                    template="/indexSinMenu.jsp">
                     <put-attribute name="titulo" value=""/>
                     <put-attribute name="cuerpo" value=""/>
          </definition>
          <definition name="login"
                                   template="/login/login.jsp">
          </definition>
          <definition name="paginaPrueba"
                    extends="plantillaBasica">
             <put-attribute name="titulo" value="Pagina de prueba"/>
             <put-attribute name="cuerpo" value="/prueba.jsp"/>
          </definition>
          <!-- Usuarios -->
          <definition name="listaUsuarios"
                                   extends="plantillaBasica">
                     <put-attribute name="titulo" value="Modificación de usuarios"/>
                     <put-attribute name="cuerpo" value="/usuario/listaUsuarios.jsp"/>
          </definition>

          <definition name="altaUsuario"
                    extends="plantillaBasica">
             <put-attribute name="titulo" value="Alta de usuario"/>
                     <put-attribute name="cuerpo" value="/usuario/altaUsuario.jsp"/>
          </definition>

          ...
</tiles-definitions>
```

Figura 33.Extracto de fichero tiles.xml.

- Web.xml [Wal11]: este fichero contiene la configuración del DispatcherServlet, que es el servlet[87] que funcionará como controlador frontal cuando se utiliza Spring MVC. Utilizando la etiqueta "servlet" se lleva a cabo esta acción. A continuación aparece la etiqueta "servlet-mapping" cuya configuración indica que dicho servlet va a ser el responsable de gestionar la totalidad de solicitudes, incluyendo las referencias a contenido estático. La siguiente etiqueta "context-param" incluye una referencia al fichero aas11-servlet.xml que determina los beans que se utilizarán. La etiqueta "listener" permite la definición de un listener que se utilizará como elemento asociado a los filtros "filter" utilizados para implementar la seguridad en la aplicación AAS11:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
        xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
  <display-name>Archetype Created Web Application</display-name>
  <servlet>
          <servlet-name>aas11</servlet-name>
          <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
          <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
          <servlet-name>aas11</servlet-name>
          <url-pattern>/</url-pattern>
  </servlet-mapping>
  <context-param>
                  <param-name>contextConfigLocation</param-name>
                  <param-value>/WEB-INF/aas11-servlet.xml</param-value>
          </context-param>
  <listener>
                  <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
  </listener>
  <!-- Spring Security -->
  <filter>
          <filter-name>springSecurityFilterChain</filter-name>
          <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
  </filter>
  <filter-mapping>
    <filter-name>springSecurityFilterChain</filter-name>
    <url-pattern>/*</url-pattern>
 </filter-mapping>
</web-app>
```

Figura 34. Fichero web.xml.

[87] Un servlet en un sentido general es un módulo que nos permite extender la funcionalidad de un servidor dotándolo de mayores capacidades.

6.3.2.3 Diseño de la capa de presentación de la aplicación

Inicialmente se elabora un diseño de la pantalla principal que utilizará la aplicación AAS11. Este diseño se utilizará como base para la definición del aspecto final de la aplicación:

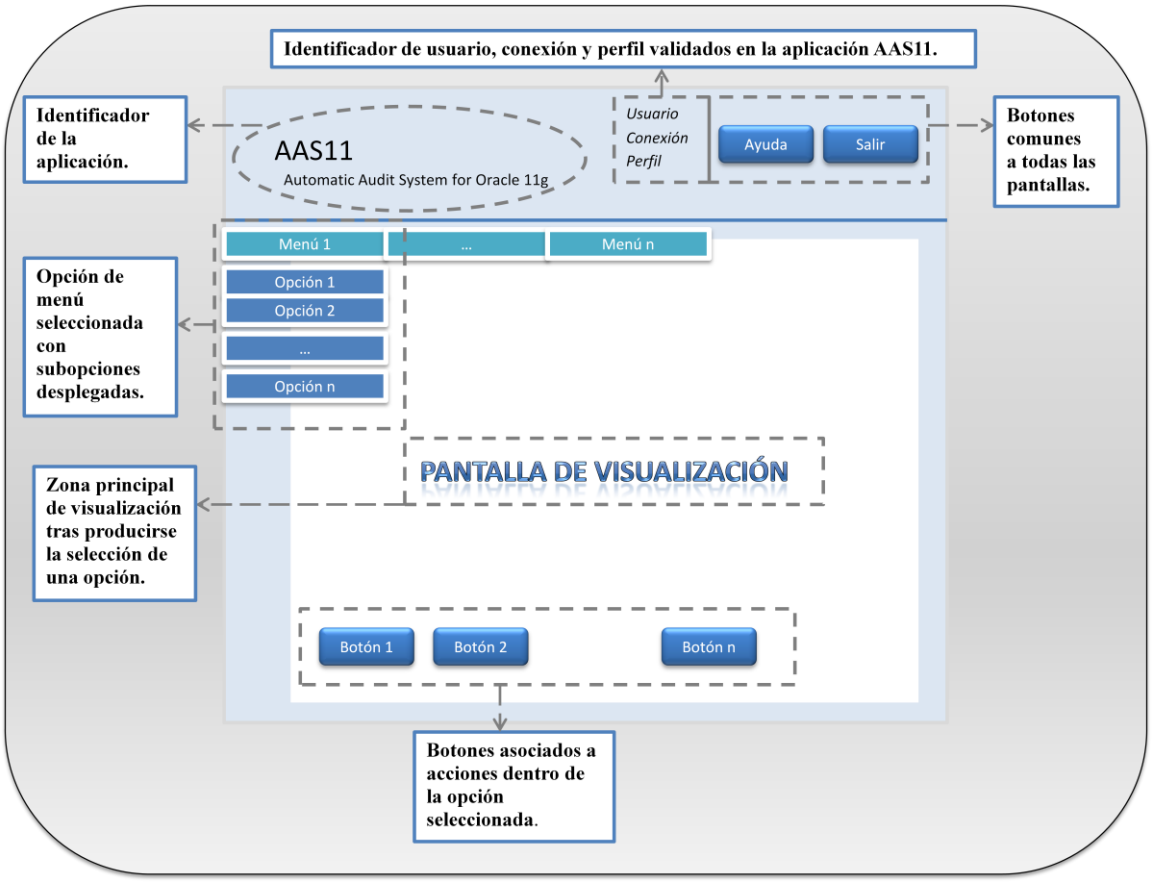


Figura 35.Diseño de pantalla principal en la aplicación AAS11.

En el diseño podemos observar los siguientes detalles:

- En la parte superior izquierda aparecerá el identificador de la aplicación.
- En la parte superior derecha aparecerán el usuario, identificador de conexión y el perfil validado en la aplicación. Adicionalmente, también aparecerán aquellos botones cuya disponibilidad debe de ser permanente.

- Debajo de los elementos anteriores se presenta el menú de la aplicación. La pulsación sobre cada una de las opciones de menú desplegará un submenú que detallará las acciones que se pueden realizar.

- La zona central de la pantalla se utilizará para representar cada una de las subopciones seleccionadas.

- Finalmente, en la parte inferior de la pantalla (dependiente de cada opción seleccionada) aparecerán una serie de botones que permitirán realizar diversas acciones sobre la opción actualmente visualizada en pantalla.

6.3.2.4 Prototipo de pantallas

Después de establecer el diseño inicial de la pantalla, se elabora un prototipo básico que permite la navegación entre dos pantallas de la aplicación. Este prototipo se utilizará como base para realizar la presentación final de la aplicación y simultáneamente, permitirá la validación de la misma.

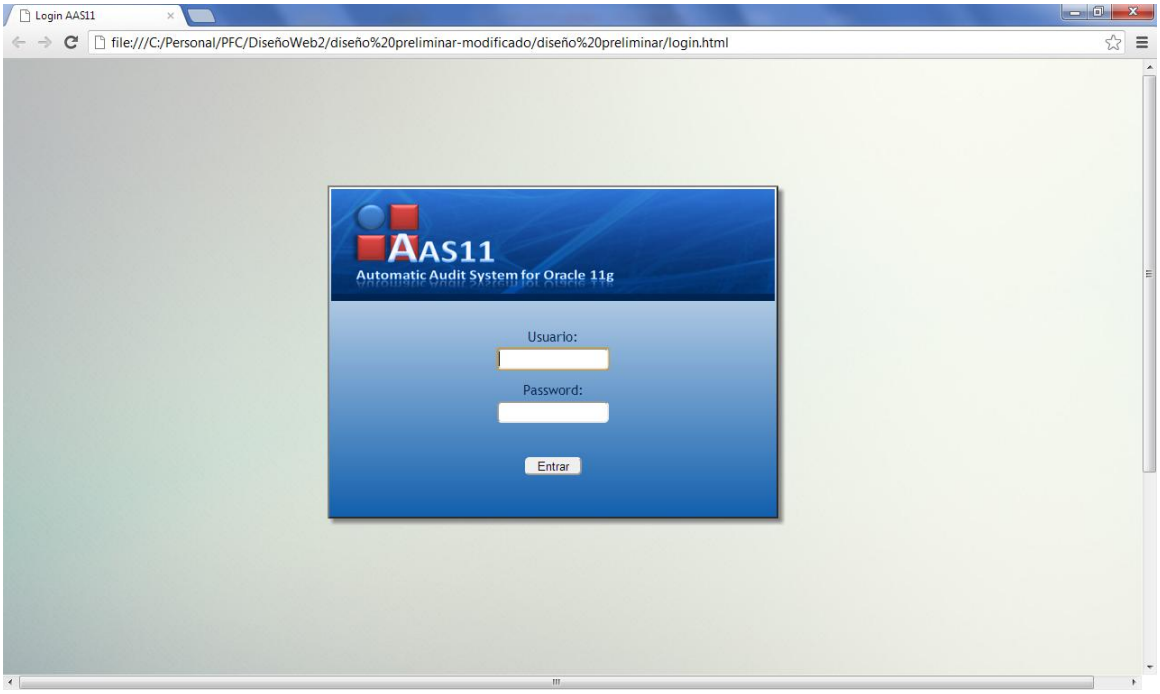


Figura 36.Pantalla de login.

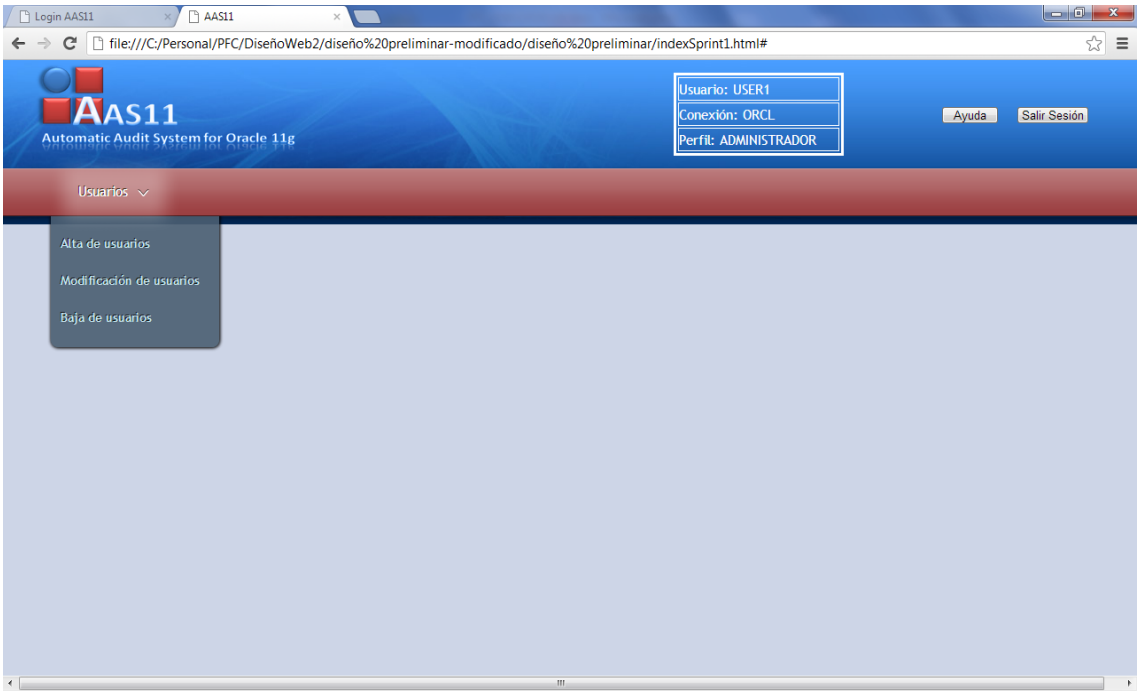


Figura 37.Pantalla principal de la aplicación con menú de usuarios.

La elaboración de este prototipo, al que progresivamente se le irán añadiendo funcionalidades, implica la utilización de las siguientes hojas de estilo[88]:

- Jquery-ui-1.9.2.custom.css: hoja de estilos asociada a la librería javascript jquery utilizada para presentar los cuadros de diálogo empleados para notificar errores en la entrada de formularios.

- Menu.css: hoja de estilos utilizada para representar aspectos generales de la aplicación.

- Styles.css: hoja de estilos aplicada a elementos que aparecen en la cabecera de la aplicación.

- StylesCont.css: hoja de estilos aplicada sobre los elementos que aparecen en la zona de visualización principal de la aplicación.

- StylesLogin.css: hoja de estilos aplicada sobre los elementos que aparecen en la pantalla de login de la aplicación.

[88] [GM01] Las hojas de estilo en casada o CSS son un estándar desarrollado por el World Wide Web Consortium (W3C) para dotar a las páginas web de mayores capacidades de visualización, sin necesidad de añadir al lenguaje HTML nuevas etiquetas de formato.

6.3.3 Tareas asociadas al requisito PB-0-002

6.3.3.1 Diagrama de casos de uso

Los diagramas de casos de uso [BRJ99] se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. Los casos de uso proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema. Además, los casos de uso ayudan a validar la arquitectura y a verificar el sistema mientras evoluciona a lo largo del desarrollo.

En la siguiente figura aparece el diagrama de casos de uso que permite representar la gestión de usuarios que se llevará a cabo en la aplicación:

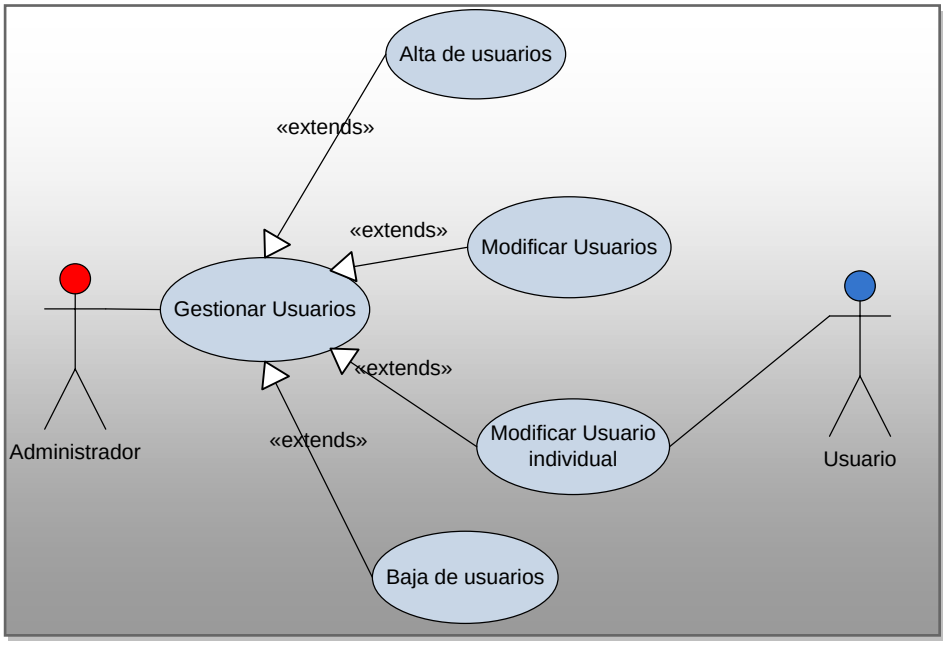


Figura 38.Diagrama de casos de uso "Gestionar Usuarios".

En este diagrama de casos de uso podemos observar como se ha representado el rol de "Administrador" y el rol de "Usuario". En el detalle de esta representación podemos ver que aquellos usuarios que dispongan del rol "Administrador" podrán llevar a cabo todas las acciones asociadas a la gestión de usuarios (altas, bajas y modificaciones). Un usuario que disponga del rol "Usuario" únicamente podrá llevar a cabo la modificación de datos sobre su propio usuario.

6.3.3.2 Generación de esquema de base de datos

Como paso previo a la construcción del modelo de datos, se procede a generar el esquema de base de datos. Para ello se establece que el esquema AAS11 se apoyará en dos Tablespaces. Cada uno de estos Tablespaces se apoyará, a su vez, en un Datafile. A continuación se representa la arquitectura de soporte utilizada en el esquema AAS11:

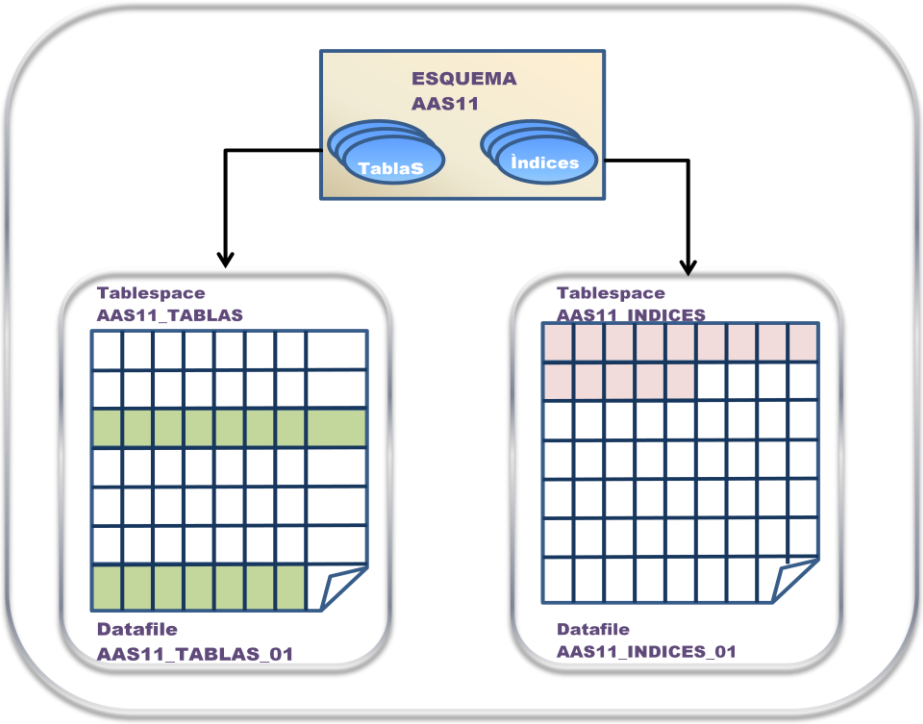


Figura 39.Esquema de base de datos asociado a usuario AAS11.

6.3.3.3 Elaboración del modelo de datos

Con la finalidad de llevar a cabo un prototipo funcional lo más rápido posible, se procederá a elaborar tanto el modelo conceptual como el modelo lógico de datos. El modelo conceptual [MPM99] se utiliza para describir aquella parte del mundo real (universo del discurso[89]) que se pretende reflejar. Este tipo de modelo deberá ser altamente semántico e independiente del sgbd en el que posteriormente se vaya a realizar la implementación de la base de datos, y también independiente del modelo lógico generado en la correspondiente fase de diseño. A diferencia del modelo conceptual, el modelo lógico representa la estructura específica utilizada por el propio sgbd y por tanto está sometido a las restricciones que imponga el mismo.

[89] Visión del mundo real bajo unos determinados objetivos [MPM99].

En el modelo lógico los conceptos que se manejan son propios del sgbd (tablas, relaciones, restricciones…) y están orientados a la representación final de los datos en la propia máquina.
Por estas razones surgen los modelos conceptuales, más semánticos y cercanos al usuario, que se utilizan para acercar la información representada en la base de datos a conceptos del mundo real.
En la siguiente figura se representa el modelo conceptual, considerando los tipos de entidad asociados a la funcionalidad "Gestionar Usuarios"[90]. En este diagrama aparecen los tipos de entidad MENUS, OPCIONES, PERFILES y USUARIOS que son utilizadas para elaborar un primer prototipo funcional de la aplicación.

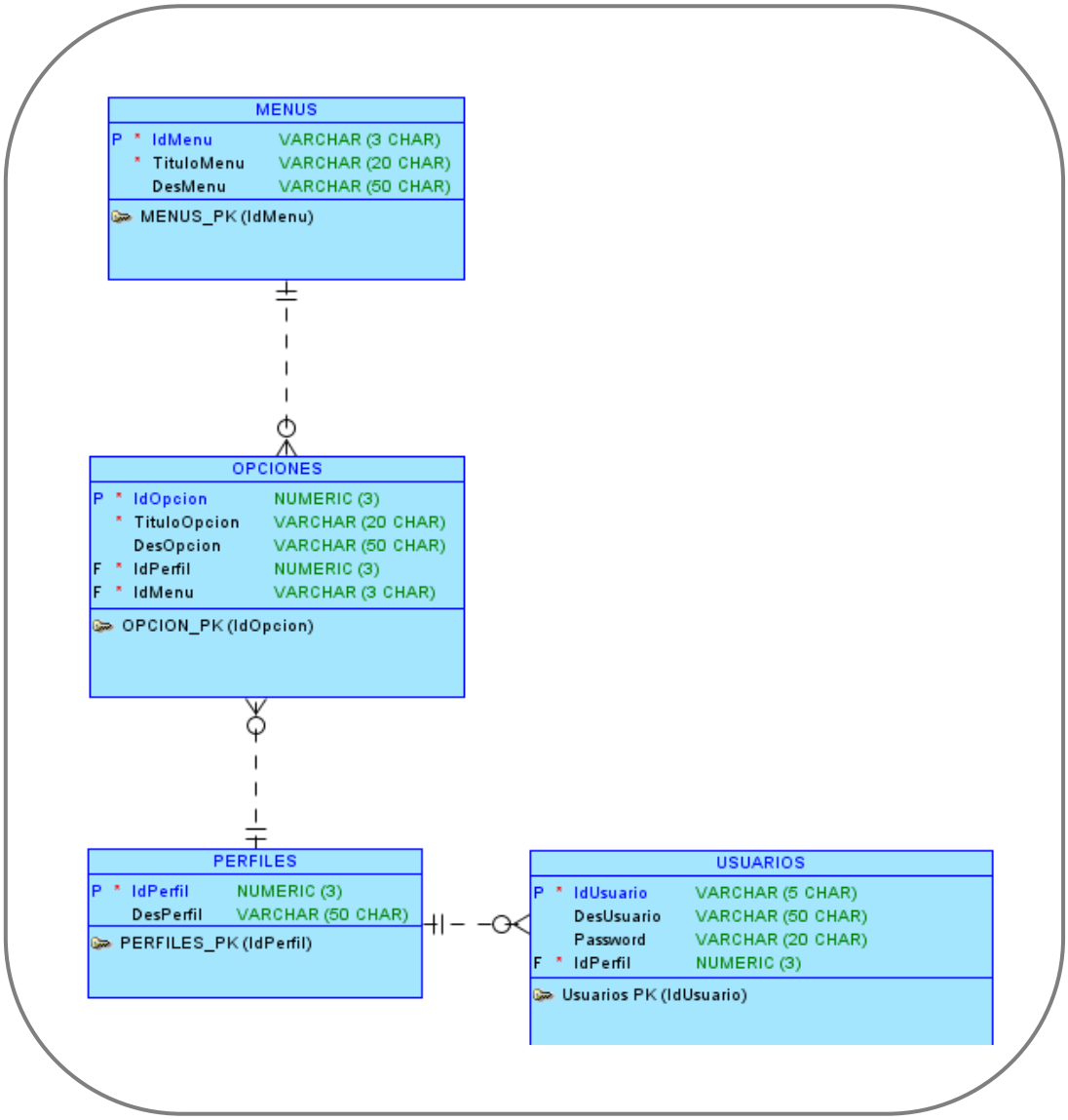


Figura 40.Modelo conceptual asociado a la Gestión de usuarios[91].

[90] La totalidad de diagramas asociados al modelo lógico se han generado con la herramienta Oracle SQL Developer Data Modeler versión: 3.1.1.703.
[91] La notación utilizada en el diagrama conceptual se denomina "Notación de Ingeniería de la Información". Existen otras notaciones disponibles como "Barker" o "Bachman".

En la siguiente figura aparecen las tablas MENUS, OPCIONES, PERFILES y USUARIOS obtenidas a partir de los tipos de entidad definidos en el modelo conceptual. En este diagrama se pueden apreciar los campos, las definiciones de clave primaria y clave ajena y los índices asociados a determinados campos.

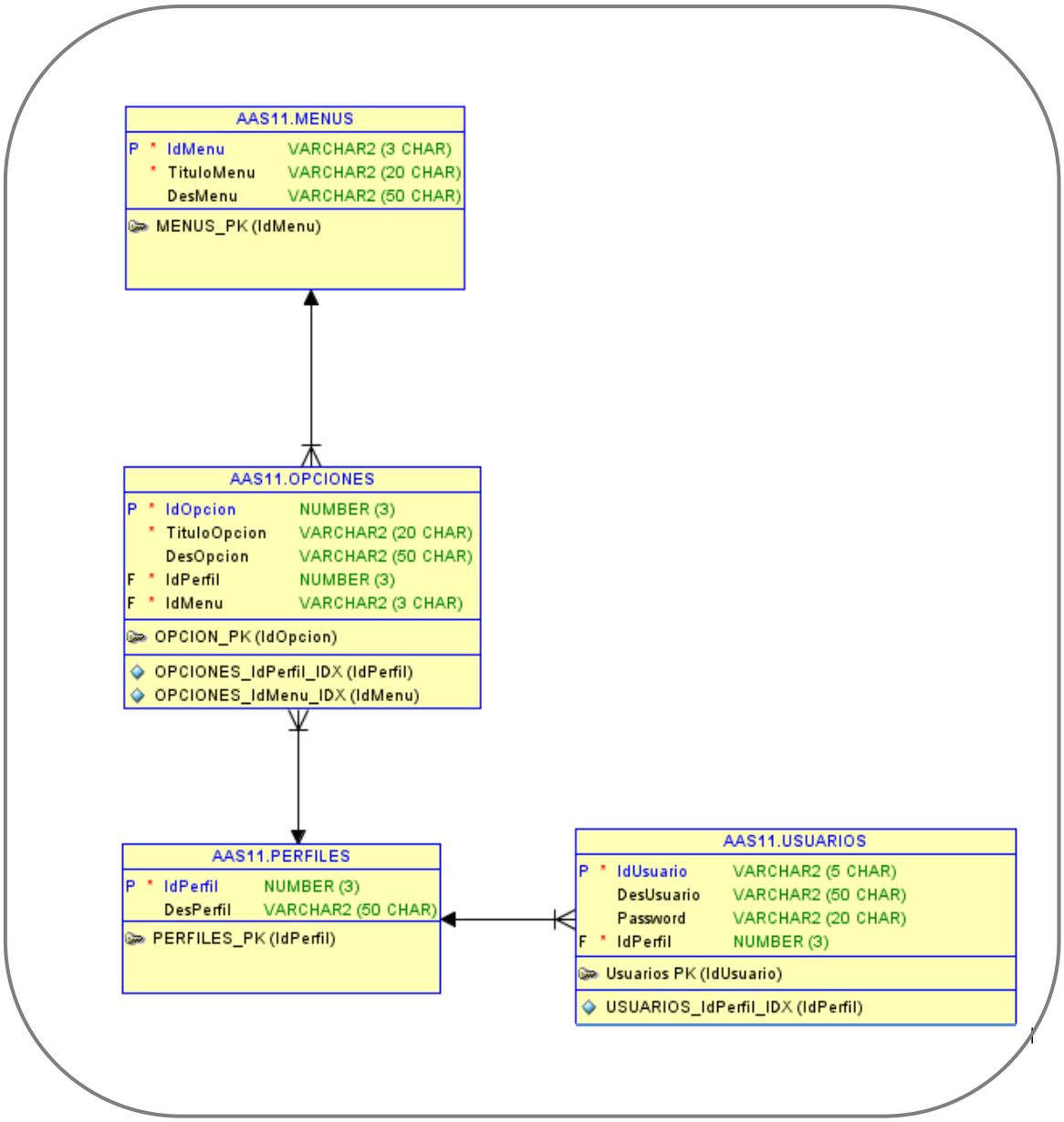


Figura 41.Modelo lógico asociado a la Gestión de usuarios.

6.3.3.4 Implementación del modelo de datos

Una vez establecidos tanto el esquema conceptual como el esquema lógico de la aplicación se procede a generar el script que permite llevar a efecto el modelo de datos planteado. La aplicación *Oracle SQL Developer Data Modeler* permite la generación automática del script utilizado para crear este modelo. En la siguiente figura aparece un extracto de parte del script asociado a la generación de la tabla PERFILES:

```
CREATE TABLE AAS11.PERFILES
(
  -- Identificador de perfil susceptible de asociarse a los usuarios del sistema
  -- AAS11.
  IdPerfil NUMBER (3)  NOT NULL ,
  -- Descripción del perfil asignable a los usuarios del sistema AAS11.
 DesPerfil VARCHAR2 (50 CHAR) ) ;

COMMENT ON COLUMN AAS11.PERFILES.IdPerfil IS 'Identificador de perfil
susceptible de asociarse a los usuarios del sistema AAS11.' ;

COMMENT ON COLUMN AAS11.PERFILES.DesPerfil IS 'Descripción del perfil
asignable a los usuarios del sistema AAS11.' ;

ALTER TABLE AAS11.PERFILES
ADD CONSTRAINT PERFILES_PK PRIMARY KEY (IdPerfil ) ;
```

Figura 42.Script de generación de la tabla PERFILES.

Este script se irá completando a medida que se vayan introduciendo elementos en la base de datos. En esta primera versión incluirá la generación de las tablas MENUS, OPCIONES, PERFILES y USUARIOS. Adicionalmente, en este script se generan las relaciones correspondientes y se crean las claves e índices necesarios.

6.3.3.4.1 Carga de datos en la base de datos

En este punto se procede a llevar a cabo la carga de datos en la propia base de datos. La carga de datos asociada a este Sprint incluye:

- Incorporación de los perfiles de “Usuario” y “Administrador”.
- Inserción de usuarios utilizados para la realización de pruebas asociadas a las operaciones de alta, baja y modificación.
- Alta de opciones y menús utilizados para la visualización de opciones asociadas a la “Gestión de usuarios” en la pantalla.

6.3.3.5 Construcción de procedimiento de Login

El diagrama de casos de uso asociado al procedimiento de login se muestra en la figura que aparece a continuación:

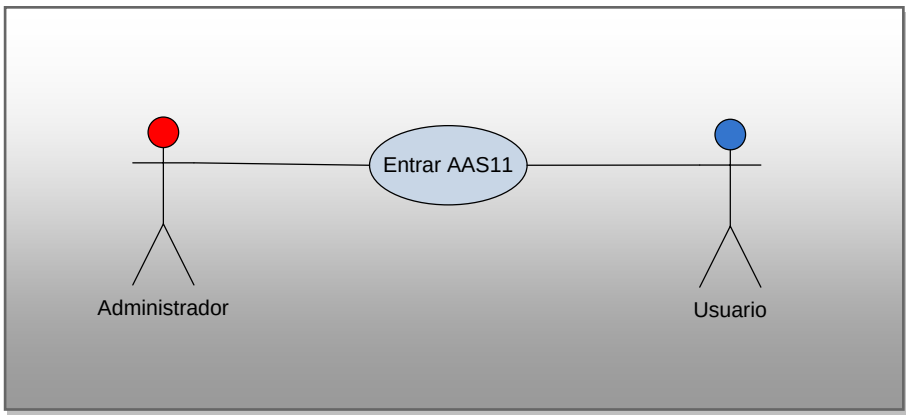


Figura 43. Diagrama de casos de uso "Entrar AAS11".

Aprovechando la construcción del procedimiento de login se introducirá la utilización del marco de trabajo Spring Security. Spring Security [Wal11] es un marco de trabajo de seguridad que proporciona seguridad declarativa para las aplicaciones basadas en Spring. Asimismo, cuenta con una solución de seguridad integral que gestiona la autenticación y la autorización, tanto a nivel de solicitud Web como a nivel de ejecución de método. Spring Security hace frente a la seguridad desde dos ángulos. Para proteger solicitudes Web y restringir el acceso al nivel de URL, utiliza servlet filters. Spring Security también puede proteger las invocaciones de método utilizando Aspect Oriented Programing (AOP[92]) de Spring, aplicando proxy sobre objetos y aplicando consejos que garanticen que el usuario cuenta con el privilegio adecuado para invocar métodos asegurados.

A la hora de aplicar Spring Security se han seguido los siguientes pasos:

- Inclusión del espacio de nombres asociado a Spring Security.
- Protección de solicitudes web aplicando filtros de servlet.
- Construcción de página de login y cierre de sesión.
- Aplicación de la seguridad con expresiones de Spring.

[92] La programación orientada a aspectos [Wal11] (en inglés: AOP Aspect Oriented Programing) se define como una técnica que promueve la separación de problemas. Elementos transversales como la administración de transacciones o la gestión de la seguridad son incluidos en módulos y aplicados de forma declarativa a los componentes que los utilizan. Esto permite que los componentes tengan mayor cohesión entre sí y puedan centrarse en sus cometidos concretos.

Con la finalidad de incluir el espacio de nombres asociado a Spring Security se ha realizado una actualización del fichero aas11-servlet.xml:

```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:jee="http://www.springframework.org/schema/jee"
       xmlns:security="http://www.springframework.org/schema/security"
       xsi:schemaLocation="http://www.springframework.org/schema/mvc
            http://www.springframework.org/schema/mvc/spring-mvc-3.1.xsd
            http://www.springframework.org/schema/context
            http://www.springframework.org/schema/context/spring-context-3.1.xsd
            http://www.springframework.org/schema/beans
            http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
            http://www.springframework.org/schema/jee
            http://www.springframework.org/schema/jee/spring-jee-3.1.xsd
            http://www.springframework.org/schema/security
            http://www.springframework.org/schema/security/spring-security-3.1.xsd">
  ...
```

Figura 44.Inclusión del espacio de nombres de Spring Security en fichero aas11-servlet.xml.

La protección de solicitudes web aplicando filtros servlet se lleva a cabo en el archivo de configuración web.xml cómo muestra la siguiente figura:

```
  ...
  <!-- Spring Security -->
<filter>
        <filter-name>springSecurityFilterChain</filter-name>
        <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
        <filter-name>springSecurityFilterChain</filter-name>
        <url-pattern>/*</url-pattern>
</filter-mapping>
  ...
```

Figura 45. Utilización de filtros de servlet sobre fichero web.xml.

En la figura anterior se puede observar como se utiliza un filtro que se apoya en una clase denominada *DelegatingFilterProxy*. Esta clase constituye un filtro de servlet especial que, por sí sólo, no desempeña ninguna funcionalidad. En su lugar, delega sobre una implementación de javax.servlet.Filter. En las líneas de código delimitadas por la etiqueta *<filter-mapping>* se establece el nombre del filtro y el patrón url que establece las URL que serán interceptadas para la aplicación de la normativa de seguridad establecidas (en este caso todas, identificadas por el patrón /*).

El siguiente paso a llevar a cabo será la construcción de una página de inicio de sesión. Para enlazar la página de inicio de sesión con el marco de trabajo de Spring Security se incluirán las siguientes líneas en el fichero de configuración aas11-servlet.xml:

```
...

<!-- Configuracion de seguridad -->
<security:global-method-security secured-annotations="enabled" pre-post-
annotations="enabled"/>
        <security:http  auto-config="true"
                use-expressions="true">
                  <security:form-login login-page="/login"
                     authentication-success-handler-ref="aas11AuthenticationUserHandler"
                          authentication-failure-url="/loginFallido"/>
                  <security:logout logout-success-url="/logout" />
        </security:http>
        <security:authentication-manager>
                  <security:authentication-provider>
                          <security:jdbc-user-service data-source-ref="dataSource"
                              users-by-username-query="select IDUSUARIO,PASSWORD,1 from
USUARIOS where IDUSUARIO = ?"
                              authorities-by-username-query="select U.IDUSUARIO,
ltrim(to_char(OP.IDOPCION, '99')) IDOPCION from PERFILES PE, MENUS ME, USUARIOS U,
OPCIONES OP where  U.IDUSUARIO = ? and OP.IDPERFIL = U.IDPERFIL and
PE.IDPERFIL=OP.IDPERFIL and ME.IDMENU=OP.IDMENU"/>
                  </security:authentication-provider>
        </security:authentication-manager>

...
```

Figura 46. Configuración de seguridad aplicada sobre el fichero aas11-servlet.xml.

En esta configuración nos centraremos en una serie de elementos para describir su utilidad. La URL empleada para la página de inicio de sesión se identifica a través del atributo *login-page*. A través del atributo *authentication-success-handler-ref* se identifica el método utilizado para realizar las acciones asociadas a la validación. En el caso de que el proceso de autenticación no haya tenido éxito, el atributo *authentication-failure-url* define la URL que se utilizará cuando se produzca un problema.

A través del atributo *users-by-username-query* contenido en el elemento *jdbc-user-service* se define la consulta utilizada para obtener el identificador de usuario y la palabra clave asociada a un determinado usuario. El atributo *authorities-by-username-query* se utiliza para establecer el listado de opciones accesibles para el usuario que está realizando la validación.

Adicionalmente, incluido en la etiqueta <*security:logout*> se utiliza el atributo logout-success-url que define la URL usada para abandonar la aplicación y provocar el cierre de sesión (funcionalidad asociada al botón SALIR). El elemento <*security: authentication-manager*> se utiliza con la finalidad de definir el procedimiento de autenticación que se ha establecido. En el caso particular de la aplicación AAS11 se utilizará un procedimiento basado en la información almacenada en la base de datos.

Una vez montada la utilización del marco de trabajo Spring Security podemos observar un ejemplo de su aplicación sobre la funcionalidad que permite dar de alta un nuevo usuario en la clase UsuarioController:

```
// Crear usuario
@PreAuthorize("hasRole('1')")
@RequestMapping("/mostrarFormularioAltaUsuario")
public ModelAndView mostrarFormularioAltaUsuarios() {
    ModelAndView modelAndView = new ModelAndView("altaUsuario");
    UsuarioForm form = new UsuarioForm();
    form.setPerfiles(this.construirMapPerfiles());
    modelAndView.addObject("command", form);
    return modelAndView;
}
```

Figura 47. Detalle del método mostrarFormularioAtlaUsuarios en la clase UsuarioController.

En la figura 43 podemos observar que, como elemento previo a la definición del método, aparece la utilización de la expresión de seguridad *hasRole* a través de la anotación *@PreAuthorize*. La utilización de esta expresión de seguridad limita el uso de la funcionalidad asociada al "alta de usuarios" a aquellos usuarios que tienen la Opción 1 disponible en su listado de opciones accesibles.

A continuación se visualiza un diagrama de clases en el que se muestra la estructura estática que representa las clases y las relaciones implicadas en la aplicación de Spring Security:

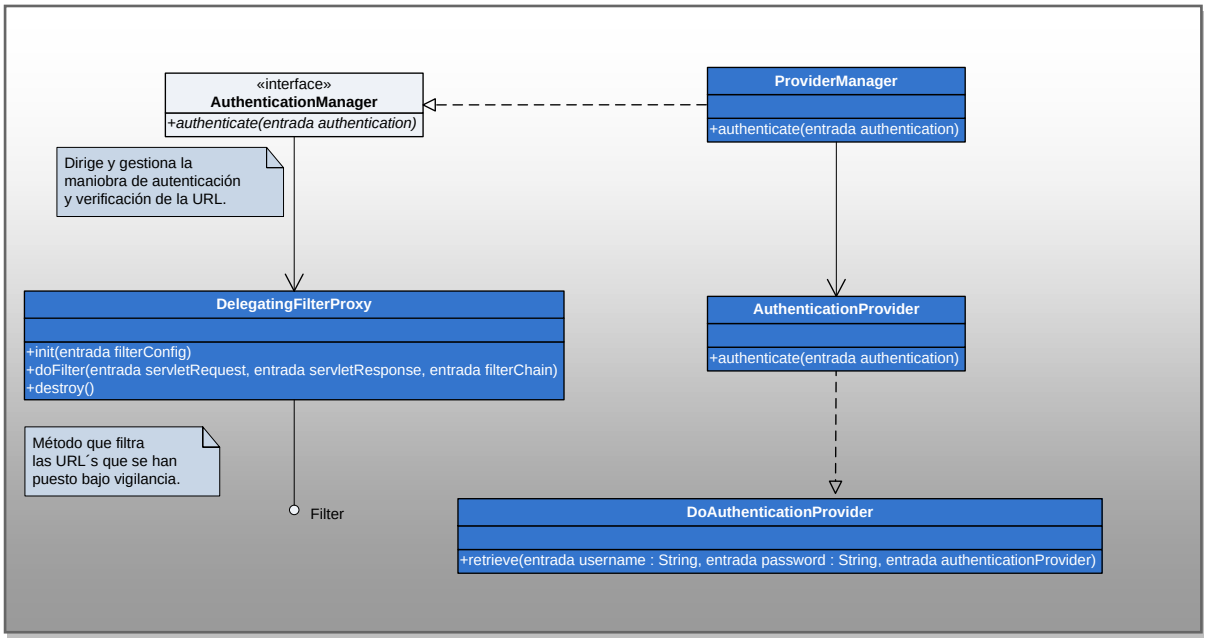


Figura 48. Diagrama de clases asociado a la utilización de Spring Security.

En el diagrama de clases de la figura anterior, las clases asociadas al marco de trabajo de Spring Security se muestran en color azul oscuro.

Con el objetivo de visualizar los aspectos dinámicos asociados a la utilización de Spring Security se ha escogido la utilización de los diagramas de secuencia. Un diagrama de secuencia [BRJ99] es un diagrama de interacción[93] que destaca la ordenación temporal de los mensajes.

En el diagrama de secuencia que aparece en la siguiente figura podemos ver la interacción entre las clases DelegatingFilterProxy, ProviderManager, DaoAuthenticationProvider y la clase controller asociada a la opción seleccionada en un escenario en el que un usuario accede a una opción tras validarse correctamente en la aplicación AAS11:

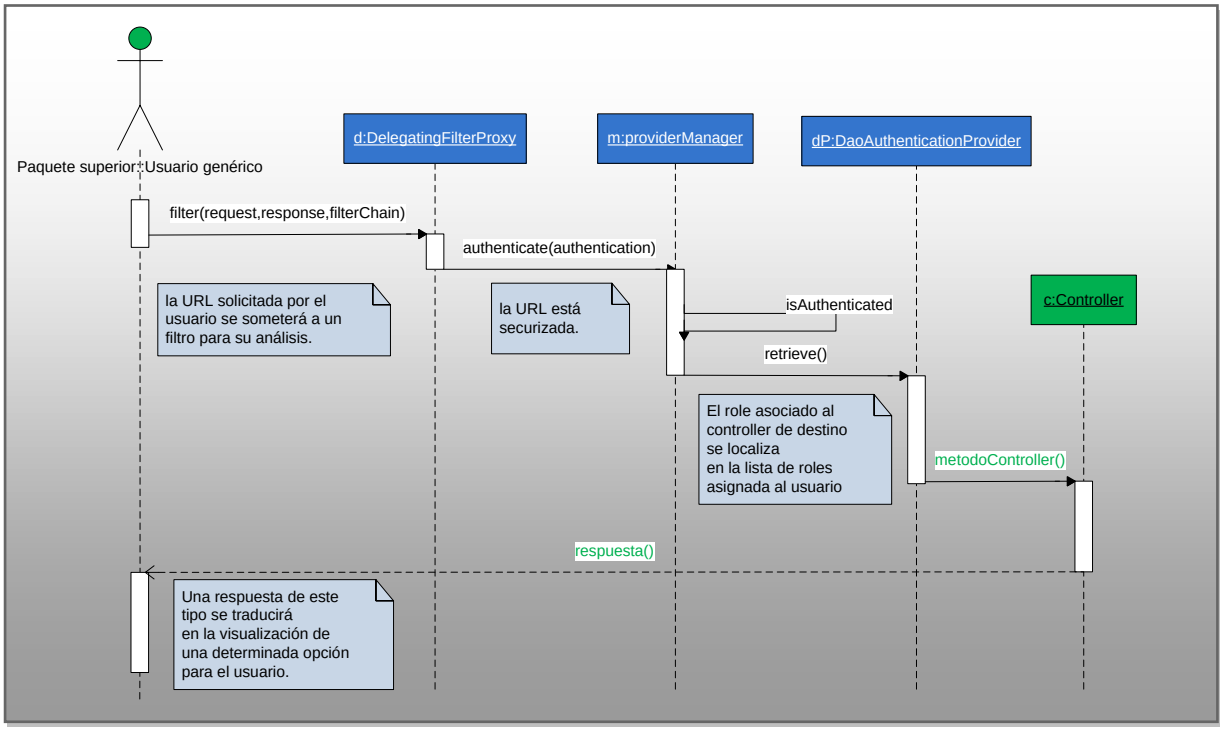


Figura 49. Diagrama de secuencia asociado a la utilización de Spring Security.

En este diagrama de secuencia, las clases en azul representan aquellas pertenecientes al marco de trabajo de Spring Security mientras que los elementos representados en verde representan objetos genéricos.

Después incluir la utilización de Spring Security, se procede a llevar a cabo la parte asociada a la vista (presentación) de la funcionalidad "Login".

[93] Los diagramas de interacción [BRJ99] se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento.

Para llevar a cabo esta parte se ha utilizado la tecnología JSP. Esta tecnología [Fro02] proporciona la capacidad de acceso a datos remotos a través de mecanismos como EJB, JDBC y RMI. Adicionalmente, permite a los desarrolladores encapsular y separar la lógica de las aplicaciones, el código Java de la presentación y el código HTML, lo que redunda en una mayor flexibilidad a la hora de crear aplicaciones y reutilización de código.

Esta separación entre lógica y presentación es la mayor de las ventajas que ofrece la tecnología JSP sobre otras arquitecturas de aplicaciones web, como los servlets o scripts CGI. Los ficheros JSP involucrados en la funcionalidad "Login" son:

- Login.jsp: fichero utilizado para presentar la pantalla de login en la aplicación. En el caso de un fallo en la introducción del usuario y la password, se visualizará en un error en la propia pantalla indicando este problema.

La respuesta HTML que se suministra al navegador tras procesar este fichero JSP en el servidor de aplicaciones se muestra en las siguientes figuras:

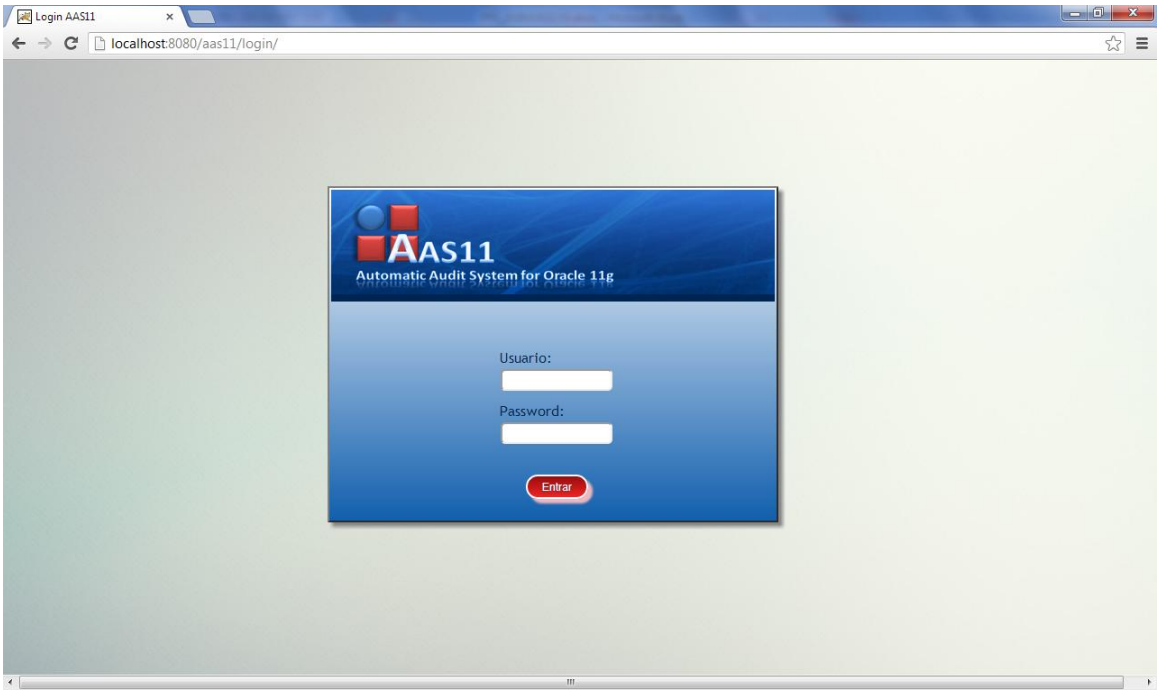


Figura 50. Pantalla de Login utilizada en la aplicación AAS11.

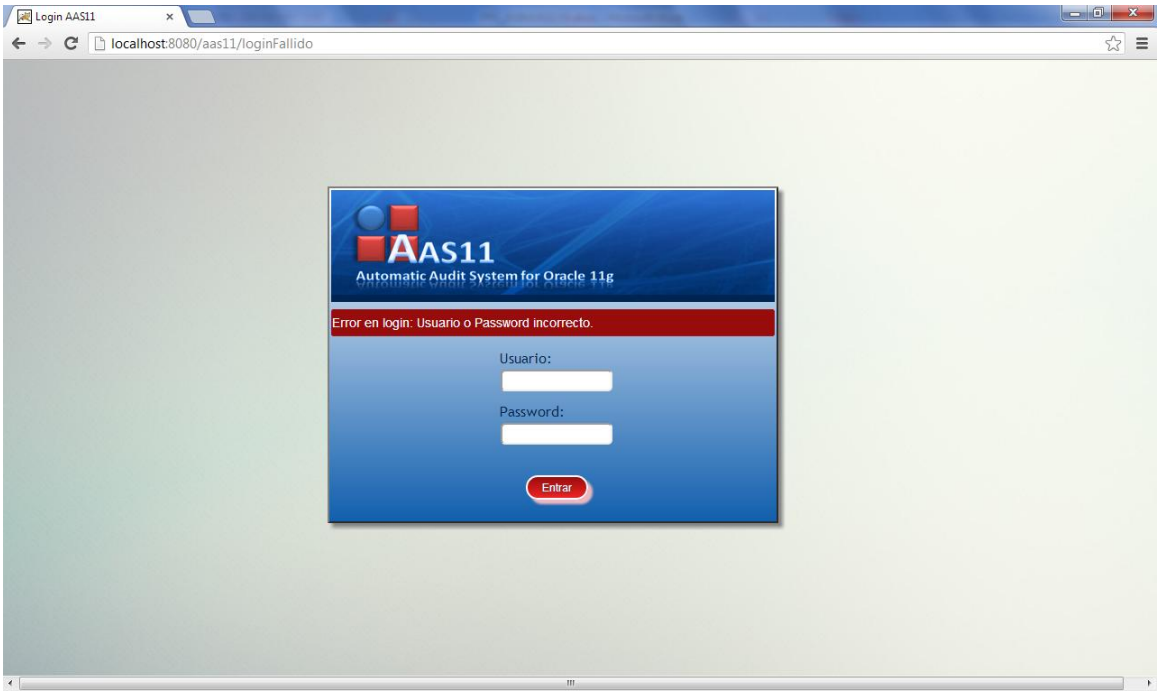


Figura 51. Error de usuario o password en pantalla de Login.

6.3.3.6 Prueba de procedimiento de Login

Con el objetivo de llevar a cabo una serie de pruebas unitarias sobre la pantalla "Login" se procede a diseñar una tabla que se utilizará para describir cada uno de los casos de prueba aplicados. En esta tabla se identifican los siguientes campos:

- Nombre del caso de prueba: identificador del caso de prueba.
- Descripción: breve descripción del caso de prueba aplicado.
- Fecha de Ejecución: fecha en la que se ha procedido a comprobar el caso de prueba.
- Responsable: responsable asignado a la ejecución de la prueba.
- Resultado: resultado de la aplicación de la prueba. Los valores posibles que puede tomar este campo son Acierto/Fallo.

Esta tabla, adicionalmente, contemplará una segunda sección que se utilizará con la finalidad de describir los posibles defectos encontrados. En esta sección se utilizan los siguientes campos:

- Defecto X: identificador del problema detectado dentro del caso de prueba tratado. La X representa un número secuencial que se irá incrementando a medida que se vayan encontrando problemas.

- Resumen: asociado a un determinado problema, breve descripción del mismo.

- Prioridad: campo que permite identificar la criticidad del problema detectado. Los valores posibles que puede tomar este campo son: Alta/Media/Baja.

- Acciones sugeridas: breve descripción del conjunto de acciones a llevar a cabo con el objetivo de corregir el problema detectado.

A continuación se muestran cada uno de los casos de prueba aplicados sobre la opción "Login":

Datos de la prueba	
Nombre del Caso de Prueba: Entrada de un usuario y clave correctos.	
Descripción: Tras la introducción de estos datos de pulsará el botón de entrar que permitirá acceder a la pantalla principal de la aplicación.	
Fecha de Ejecución: 21-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 26.Prueba unitaria de entrada de usuario y clave correctos.

Datos de la prueba	
Nombre del Caso de Prueba: Entrada de un usuario y/o clave incorrectos.	
Descripción: La entrada de estos datos producirá la visualización de un mensaje en la pantalla indicando que el usuario introducido y/o la clave introducida son incorrectos.	
Fecha de Ejecución: 21-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 27.Prueba unitaria de entrada de usuario y/o clave incorrectos.

6.3.3.7 Construcción de opción de Alta de usuarios

Como paso previo a la exposición de los diferentes elementos de diseño que se han utilizado para la realización de la aplicación AAS11, se realizará un resumen de los patrones de diseño[94] empleados como base:

- Patrón Modelo-Vista-Controlador (MVC) [Pav08] : este patrón fue originalmente descrito para el lenguaje de programación Smalltalk en 1979 y es utilizado en una amplia variedad de frameworks. Establece una separación clara entre el modelo (representación de los datos), varias vistas (utilizadas para representar el aspecto de la aplicación) y los controladores (elementos utilizados para coordinar el resto de componentes). Esta separación de aspectos de una aplicación dota de mucha flexibilidad a los desarrolladores. Una aplicación directa de este patrón puede observarse en el proyecto AAS11 a través de la utilización de distintos componentes y clases:
  - Controladores: clases controller utilizadas en el desarrollo de la aplicación.
  - Vista: ficheros jsp´s y clases form utilizadas para el intercambio de información con las vistas.
  - Modelo: clases DAO utilizadas para llevar a cabo la representación de los datos.

- Patrón Tansfer Object (TO) [Bel09]: este patrón se usa para representar los objetos de transferencia, que son los que se utilizan para el intercambio de información con la base de datos. Anteriormente conocido como Value Object (VO). Los atributos asociados a las clases que implementan este patrón se definen como privados y se utilizan métodos getter and setter para acceder y modificar el valor de los atributos. El ejemplo de aplicación dentro del proyecto AAS11 estaría constituido por las clases de tipo TO.

- Patrón Data Access Object (DAO) [Bel09]: este patrón permite la definición de componentes software que suministra una interfaz común entre la aplicación y uno o más repositorios de almacenamiento. La finalidad de la aplicación de este patrón de diseño es:

  - Abstraer y encapsular los accesos al correspondiente repositorio de datos.
  - Gestionar las conexiones a dicho repositorio.
  - Gestionar los datos almacenados en el repositorio utilizado.

  El ejemplo de aplicación dentro del proyecto AAS11 sería la utilización de clases del tipo DAO para el acceso directo a los datos.

[94] Un patrón de diseño [Bel09] es una solución efectiva a un problema de diseño que cumple las siguientes características:
- Efectivo: ha valido para resolver el problema en diseños anteriores.
- Reutilizable: La solución es aplicable a problemas similares.

El diagrama de clases asociado a la funcionalidad de "Alta de Usuarios" se muestra en la siguiente figura:

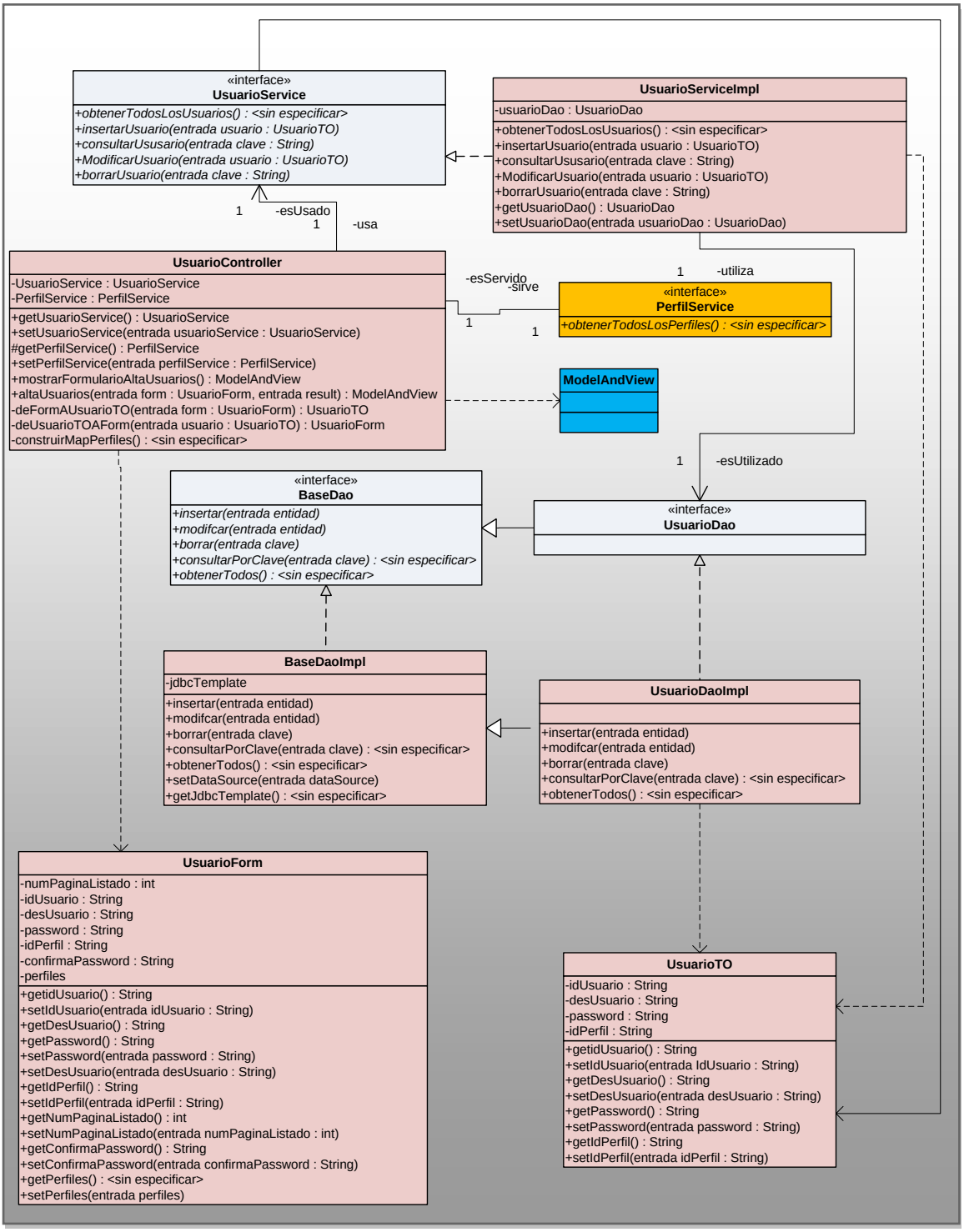


Figura 52. Diagrama de clases asociado a "Alta de Usuarios".

En este diagrama de clases aparecen la totalidad de las clases involucradas en la gestión completa de usuarios. Estas clases se han implementado de forma paralela a la implementación de la opción de alta, de tal forma que estarán disponibles tanto para la implementación de la opción de modificación, como para la implementación de la opción de baja. El color de la interfaz PerfilService indica que en este instante del desarrollo la interfaz se ha identificado como necesaria aunque todavía no ha sido implementada. El color de la clase ModelAndView[95] indica que se trata de una clase que no ha sido implementada por el desarrollador. A la hora de interpretar el diagrama de clases se expondrá el orden de implementación de las mismas junto con una breve explicación del cometido de cada una de ellas:

1. UsuarioTO: clase utilizada para el intercambio de información con la base de datos. Los atributos que se definen en esta clase representan un usuario y los métodos implementados permiten el acceso a dichos atributos.

2. UsuarioDaoImpl: clase en la que se implementan los métodos asociados a las acciones de actualización y consulta sobre la base de datos (inserción, modificación, borrado, consulta por clave y obtención de la totalidad de usuarios). La totalidad de clases DAO implementadas extenderán la clase BaseDao, en la que se definen los métodos que permiten la obtención y manipulación de los conjuntos de datos procedentes de la base de datos. Tanto sobre la clase BaseDao, como sobre la clase UsuarioDaoImpl se definen interfaces. La interfaz asociada a la clase UsuarioDaoImpl extiende la interfaz BaseDao.

3. UsuarioServiceImpl: clase utilizada para definir los servicios involucrados en la gestión de usuarios. Sobre esta clase se define una interfaz que será utilizada por la clase conroller correspondiente. Estos servicios son:

    - Obtención de una lista con todos los usuarios.
    - Inserción de un usuario.
    - Modificación de un usuario.
    - Consulta de los datos de un usuario a través de su identificador.
    - Borrado de un usuario.

4. UsuarioForm: clase que representa el conjunto de información intercambiada en los formularios utilizados en la gestión de usuarios.

5. UsuarioController: esta clase se utilizará como elemento coordinador, invocando las acciones correspondientes para llevar a cabo las funcionalidades requeridas. En este caso, se han definido los métodos privados correspondientes a la transformación de datos entre los objetos pertenecientes a la clase TO y los objetos pertenecientes a la clase FORM y los métodos que permiten la realización de las acciones de alta de usuarios.

[95] Clase perteneciente al framework de Spring contenida en el paquete *org.springframework.web.servlet*.

En las siguientes tablas aparece una breve descripción de la funcionalidad implementada en cada uno de los métodos pertenecientes a estas clases:

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioTO	getIdUsuario	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de usuario.
	setIdUsuario	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de usuario
	getDesUsuario	**Entrada:** No **Salida:** String	Método que permite la obtención de la descripción de usuario.
	setDesUsuario	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la descripción de usuario.
	getPassword	**Entrada:** No **Salida:** String	Método que permite la obtención de la palabra clave asignada al usuario.
	setPassword	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la palabra clave asignada al usuario.
	getIdPerfil	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de perfil asignado al usuario.
	setIdPerfil	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de perfil asignado al usuario.
UsuarioDaoImpl	insertar	**Entrada:** UsuarioTO **Salida:** No	Método que permite llevar a cabo la inserción en la base de datos de un nuevo usuario.
	modificar	**Entrada:** UsuarioTO **Salida:** String	Método que permite llevar a cabo la modificación en la base de datos de los datos asociados a un determinado usuario.
	borrar	**Entrada:** String **Salida:** No	Método que permite llevar a cabo el borrado en la base de datos de un usuario al completo.
	consultarPorClave	**Entrada:** String **Salida:** UsuarioTO	Método que permite llevar a cabo la consulta de los datos de un usuario a partir de su identificador.
	obtenerTodos	**Entrada:** No **Salida:** List<UsuarioTO>	Método que permite obtener la totalidad de usuarios almacenados en la base de datos.

Tabla 28.Tabla de descripción de métodos asociados a las clases UsuarioTO y UsuarioDaoImpl.

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioServiceImpl	getUsuarioDao	**Entrada:** No **Salida:** UsuarioDao	Método que permite obtener el valor del atributo usuarioDao.
	setUsuarioDao	**Entrada:** UsuarioDao **Salida:** No	Método que permite establecer el valor del atributo usuarioDao.
	obtenerTodosLosUsuarios	**Entrada:** No **Salida:** List<UsuarioTO>	Método que permite obtener una lista que contendrá el detalle de los datos asociados a la totalidad de usuarios almacenados.
	insertarUsuario	**Entrada:** UsuarioTO **Salida:** No	Método que suministra el servicio que permite llevar a cabo la inserción de un usuario.
	consultarUsuario	**Entrada:** String **Salida:** UsuarioTO	Método que suministra el servicio que permite llevar a cabo la consulta de un usuario. El parámetro de entrada se corresponde con el identificador de un usuario.
	modificarUsuario	**Entrada:** UsuarioTO **Salida:** No	Método que suministra el servicio que permite llevar a cabo la modificación de un usuario.
	borrarUsuario	**Entrada:** String **Salida:** No	Método que suministra el servicio que permite llevar a cabo la eliminación de un usuario. El parámetro de entrada se corresponde con el identificador de un usuario.

Tabla 29.Tabla de descripción de métodos asociados a la clase UsuarioServiceImpl.

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioForm	getNumPaginaListado	**Entrada:** No **Salida:** int	Método que permite obtener el valor del atributo numPaginaListado que será utilizado para implementar la funcionalidad de paginación.
	setNumPaginaListado	**Entrada:** int **Salida:** No	Método que permite establecer el valor del atributo numPaginaListado que será utilizado para implementar la funcionalidad de paginación.
	getIdUsuario	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de usuario.
	setIdUsuario	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de usuario.
	getDesUsuario	**Entrada:** No **Salida:** String	Método que permite la obtención de la descripción de usuario.
	setDesUsuario	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la descripción de usuario.
	getPassword	**Entrada:** No **Salida:** String	Método que permite la obtención de la palabra clave asignada al usuario.
	setPassword	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la palabra clave asignada al usuario.
	getConfirmaPassword	**Entrada:** No **Salida:** String	Método que permite la obtención del atributo confirmaPassword, que permitirá realizar una validación sobre la password originalmente introducida.
	setConfirmaPassword	**Entrada:** String **Salida:** No	Método que permite establecer el valor del atributo confirmaPassword, que permitirá realizar una validación sobre la password originalmente introducida.
	getIdPerfil	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de perfil asignado al usuario.
	setIdPerfil	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de perfil asignado al usuario.

Tabla 30.Tabla de descripción de métodos asociados a la clase UsuarioForm.

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioController	getUsuarioService	**Entrada:** No **Salida:** UsuarioService	Método que permite obtener el valor del atributo usuarioService, utilizado para acceder a las funcionalidades implementadas por la clase UsuarioService.
	setUsuarioService	**Entrada:** UsuarioService **Salida:** No	Método que permite establecer el valor del atributo usuarioService, utilizado para acceder a las funcionalidades implementadas por la clase UsuarioService.
	getPerfilService	**Entrada:** No **Salida:** PerfilService	Método que permite obtener el valor del atributo perfilService, utilizado para acceder a las funcionalidades implementadas por la clase PerfilService (información de perfiles).
	setPerfilService	**Entrada:** PerfilService **Salida:** No	Método que permite establecer el valor del atributo perfilService, utilizado para acceder a las funcionalidades implementadas por la clase PerfilService (información de perfiles).
	mostrarFormularioAltaUsuarios	**Entrada:** No **Salida:** No	Método que lleva a cabo la invocación al formulario de alta de usuarios.
	altaUsuarios	**Entrada:** UsuarioForm, BindingResult **Salida:** No	Método que invoca a los métodos que llevan a cabo el alta en la base de datos del usuario.
	deFormAUsuarioTO	**Entrada:** UsuarioForm **Salida:** UsuarioTO	Método que permite la transformación de objetos de tipo UsuarioForm a objetos de tipo UsuarioTO.
	deUsuarioTOAForm	**Entrada:** UsuarioTO **Salida:** UsuarioForm	Método que permite la transformación de objetos de tipo UsuarioTO a objetos de tipo UsuarioForm.
	cosntruirMapPerfiles	**Entrada:** No **Salida:** Map<String,String>	Método que permite construir el mapa de perfiles de la aplicación.

Tabla 31.Tabla de descripción de métodos asociados a la clase UsuarioController.

En la siguiente figura podemos observar el diagrama de secuencia correspondiente a la pulsación por parte de un usuario, con privilegios de "Administrador", sobre la opción "Alta de Usuarios":

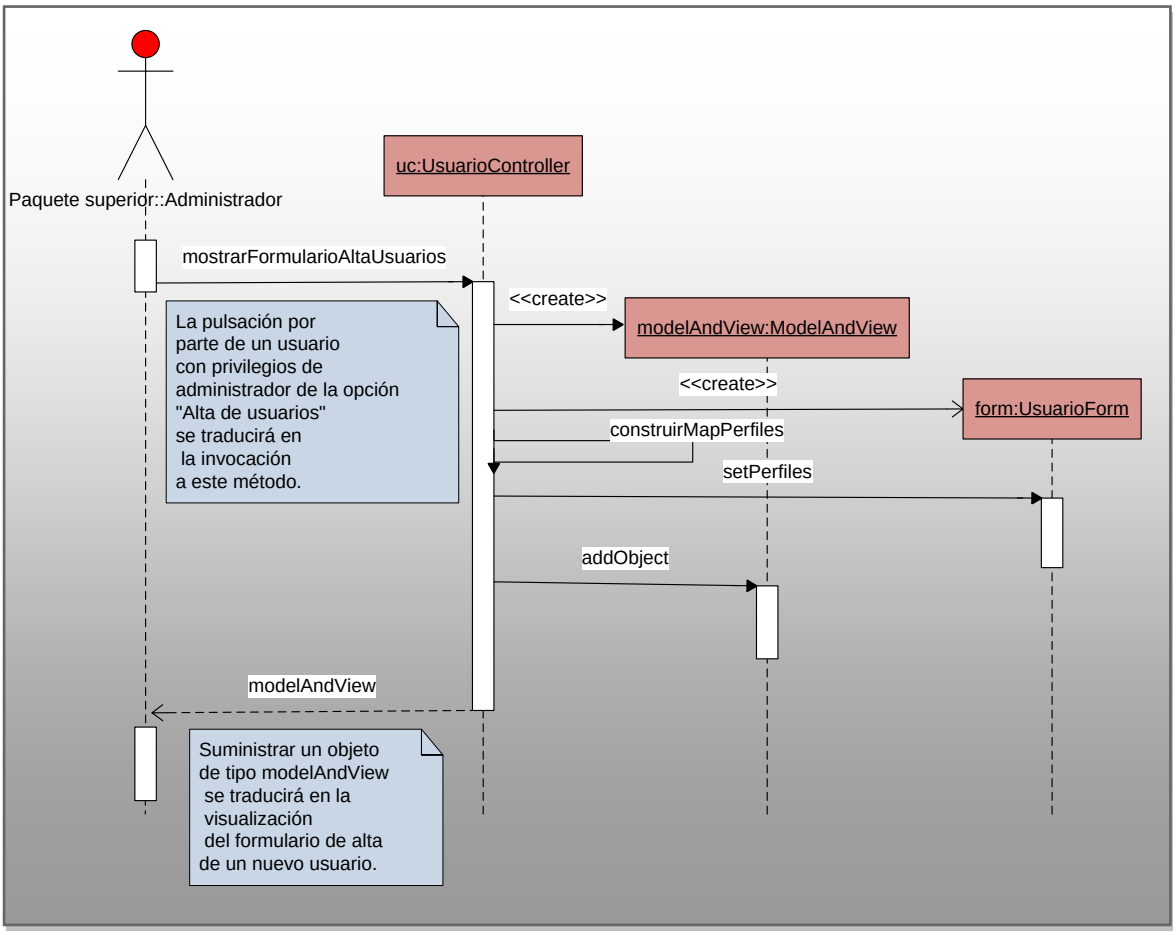


Figura 53. Diagrama de secuencia asociado a la pulsación de la opción "Alta de usuarios".

El resultado de la ejecución asociada a este diagrama de secuencia será la visualización del formulario de "Alta de usuarios" en el que un usuario con privilegios de "Administrador" podrá llevar a cabo la acción de incorporación de un nuevo usuario. El diagrama de secuencia que se muestra en la siguiente figura permite visualizar el proceso que se lleva a cabo tras la introducción de datos en el formulario de "Alta de usuarios":

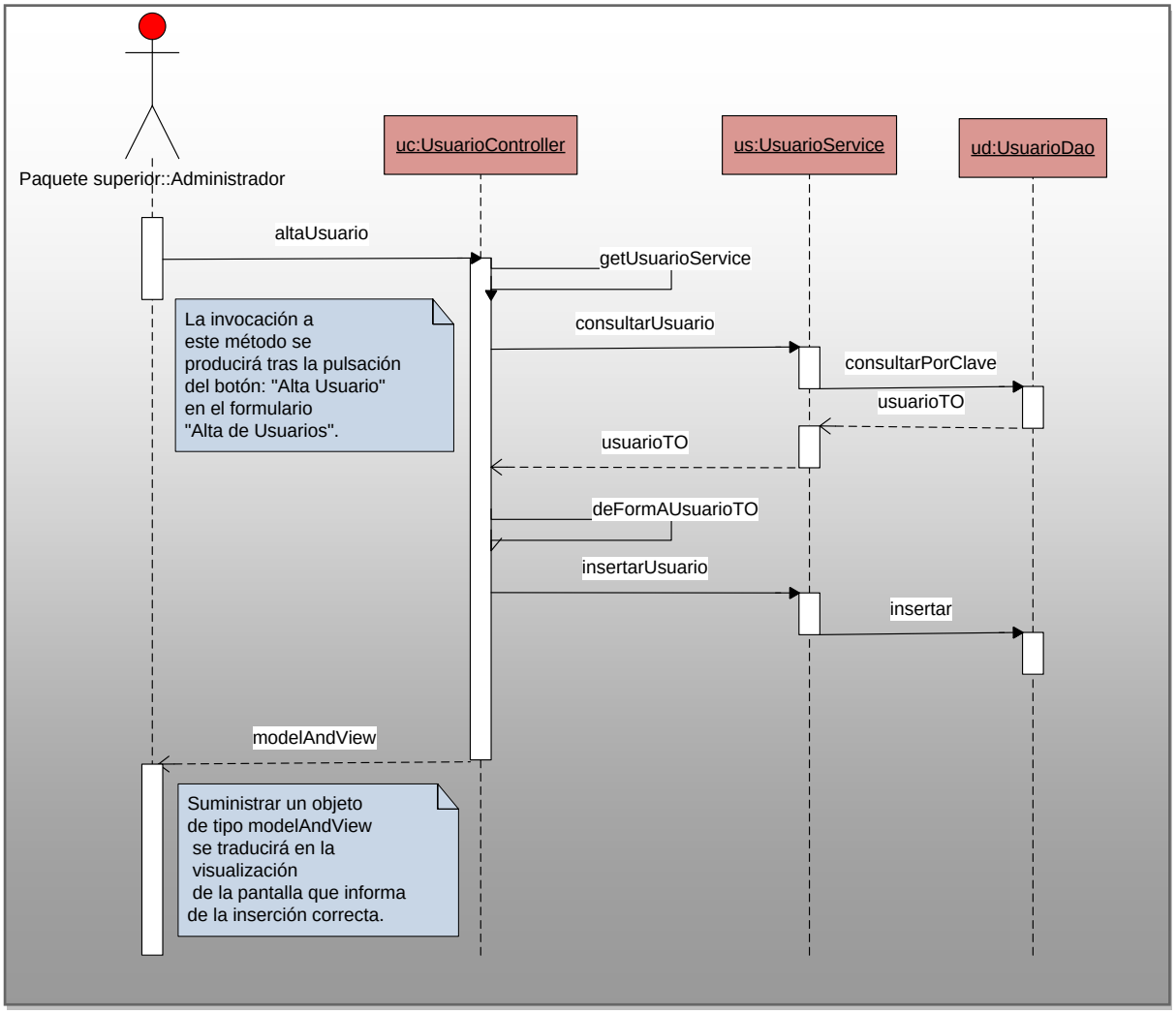


Figura 54. Diagrama de secuencia asociado a la realización de un alta tras rellenar el formulario de "Alta de usuarios".

El diagrama de secuencia de la figura superior representa un escenario en el que un usuario con permisos de "Administrador" lleva a cabo con éxito la inserción de un nuevo usuario. Los primeros intercambios de mensajes que se representan en el diagrama de secuencia se utilizan para descartar que el usuario que se está dando de alta exista actualmente en la base de datos. Una vez que se descarta este hecho, se procede a insertar el nuevo usuario.

Después de implementar las clases que gestionan el modelo de datos, la lógica de negocio y el control de acciones disponibles, se procede a llevar a cabo la parte asociada a la vista (presentación) de la funcionalidad "Alta de usuarios". Los ficheros JSP involucrados en la funcionalidad "Alta de usuarios" son:

- Index.jsp: fichero utilizado para presentar la cabecera principal de la aplicación AAS11. Esta cabecera incluye los datos del usuario que se ha validado tras realizar la acción, los botones de ayuda y logout y el menú principal de la aplicación que incluye la totalidad de opciones disponibles.

- IndexSinMenu.jsp: fichero utilizado para presentar la cabecera de la aplicación en pantallas en las que no se pondrán a disposición de los usuarios las opciones de menú de la aplicación (pantallas en las que se visualiza un mensaje informativo como resultado de una acción llevada a cabo por el usuario).

- Inicial.jsp y pantallaInicial.jsp: ficheros utilizados para representar la pantalla inicial y sin contenido de la aplicación cuando no se ha seleccionado ninguna opción del menú.

- AltaUsuario.jsp: fichero utilizado para representar el formulario de solicitud de datos para realizar el alta de un nuevo usuario.

- usuarioInsertadoCorrectamente.jsp: fichero que permite la visualización de un mensaje informativo tras la realización de la acción de alta.

La respuesta HTML que se suministra al navegador tras procesar estos ficheros JSP en el servidor de aplicaciones se muestra en las siguientes figuras:

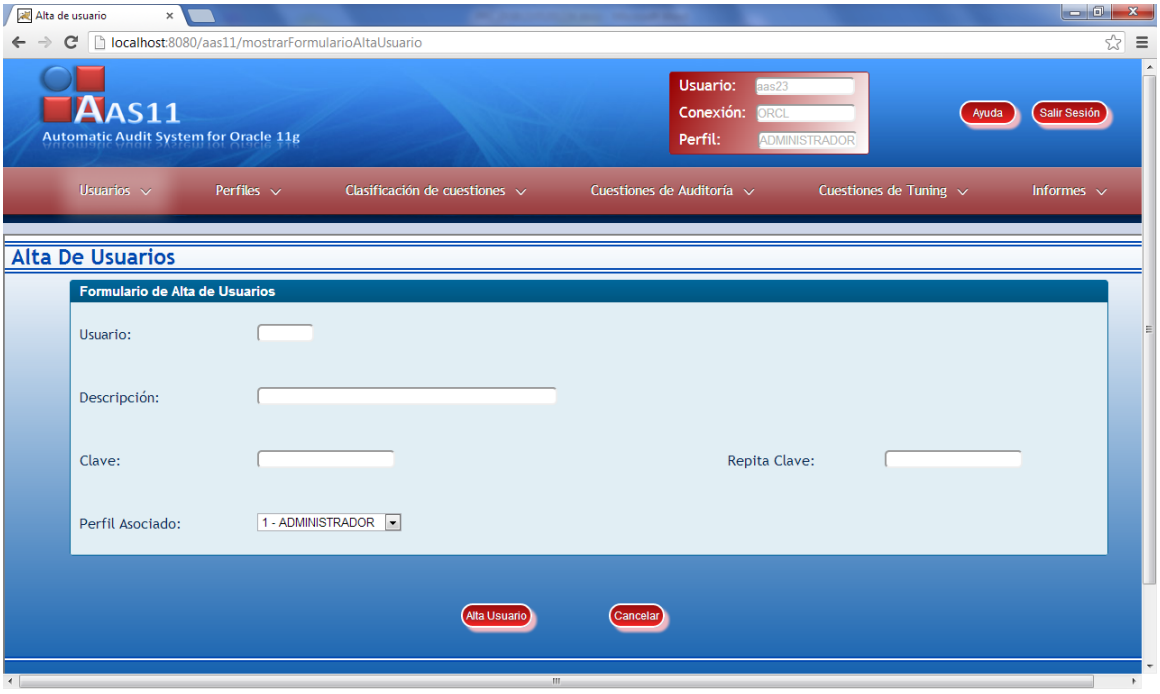


Figura 55. Presentación del formulario de Alta de Usuarios.

En la parte superior de esta figura se puede observar la interpretación del fichero index.jsp y la del fichero AltaUsuario.jsp. Gracias a la utilización de Tiles estos ficheros pueden aparecer combinados en una única pantalla.

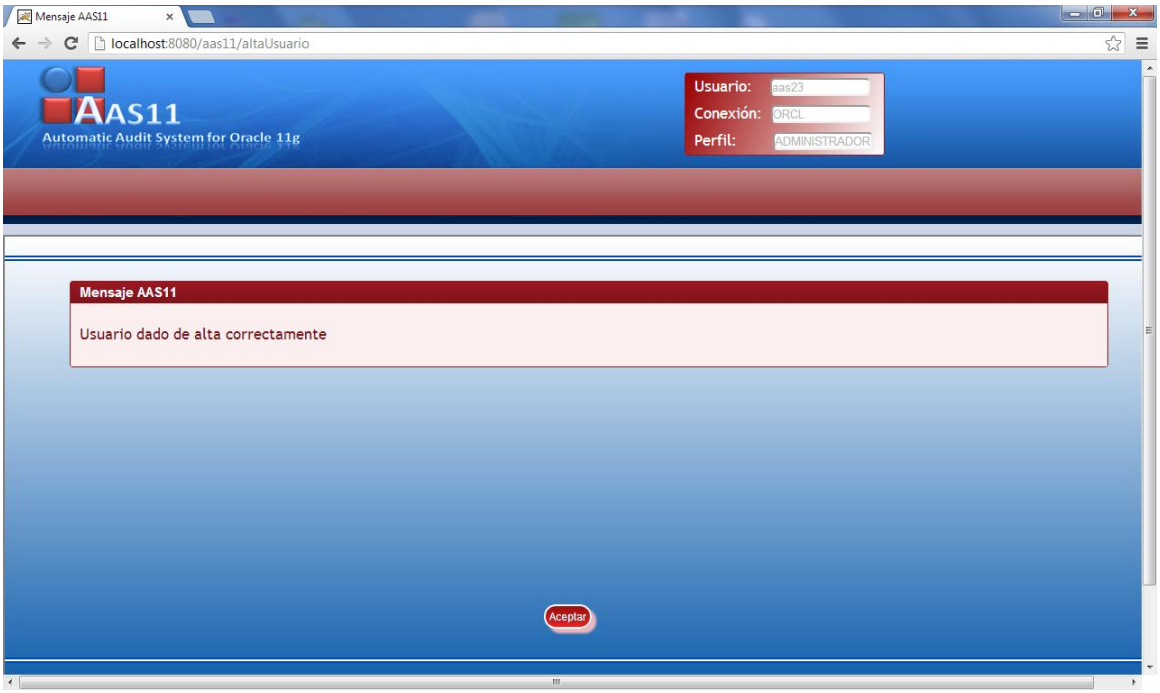


Figura 56. Presentación de mensaje tras la realización de un alta de usuario.

En la figura 56 se puede observar la interpretación del fichero indexSinMenu.jsp y la del fichero usuarioInsertadoCorrectamente.jsp.

6.3.3.8 Prueba de opción de Alta de usuarios

A continuación se muestran cada uno de los casos de prueba aplicados sobre la opción "Alta de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: selección de la opción "Alta de usuarios".	
Descripción: después de llevar a cabo la selección de la opción "Alta de usuarios" se debe mostrar un formulario en el que se soliciten la totalidad de datos asociados a la incorporación de un nuevo usuario.	
Fecha de Ejecución: 24-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 32.Prueba unitaria selección de la opción "Alta de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: visualización del formulario "Alta de usuarios".	
Descripción: La visualización del formulario "Alta de usuarios" debe incluir los campos: usuario, descripción, clave, repita clave y perfil asociado. Estos campos deben tener la dimensión correcta y en el caso del campo perfil asociado mostrar un desplegable con las opciones posibles.	
Fecha de Ejecución: 24-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 33.Prueba unitaria visualización del formulario "Alta de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: inserción correcta utilizando formulario "Alta de usuarios".	
Descripción: se suministrarán datos correctos sobre la totalidad de los campos pertenecientes al formulario "Alta de usuarios". Tras la pulsación del botón "Alta usuario" se visualizará una pantalla indicando que la inserción ha sido correcta.	
Fecha de Ejecución: 24-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja.
	Acciones sugeridas:

Tabla 34.Prueba unitaria inserción correcta utilizando formulario "Alta de usuarios".

6.3.3.9 Construcción de opción de Modificación de usuarios

El desarrollo de la opción "Alta de usuarios" se ha utilizado para implementar la infraestructura necesaria para llevar a cabo la gestión integral de usuarios. La opción de "Modificación de usuarios" se apoyará también en esta infraestructura de tal forma que la implementación de esta opción únicamente requerirá la implementación de una serie de métodos sobre la clase UsuarioController. En la siguiente figura se muestra la clase UsuarioController con los nuevos métodos añadidos:

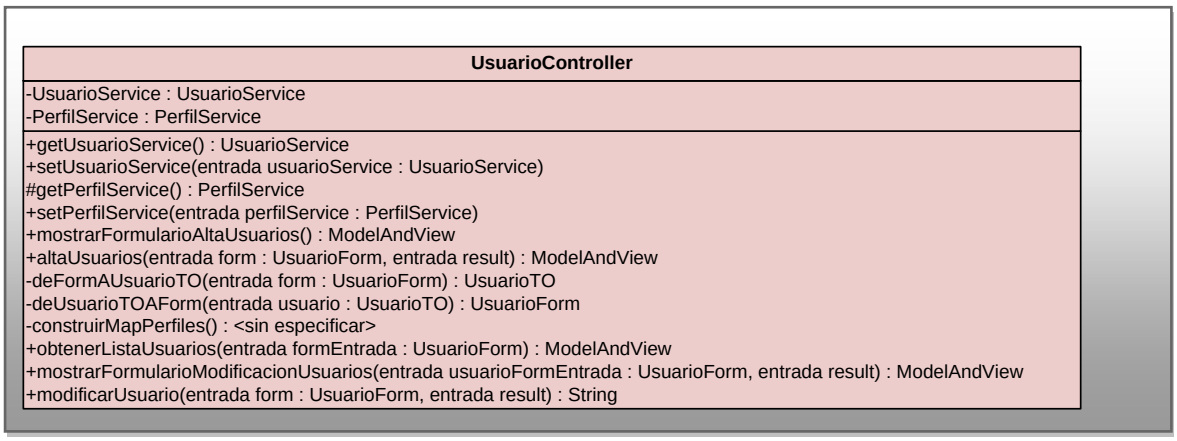


Figura 57. Clase UsuarioController con métodos asociados a la opción "Modificación de usuarios".

Los métodos asociados a la opción "Modificación de usuarios" son:

- obtenerListaUsuarios.
- mostrarFormularioModificacionUsuarios.
- modificarUsuario.

En la siguiente tabla se realiza una descripción de los métodos añadidos, asociados a la funcionalidad "Modificación usuarios":

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioController	obtenerListaUsuarios	**Entrada:** UsuarioForm **Salida:** ModelAndView	Método que permite obtener un listado con la totalidad de usuarios almacenados. A partir de este listado el usuario podrá seleccionar un usuario para llevar a cabo su modificación.
	mostrarFormularioModificacionUsuarios	**Entrada:** UsuarioForm BindingResult **Salida:** No	Método que permite mostrar un formulario con la totalidad de los datos asociados a un usuario para permitir su modificación. El único dato sobre el que no se permitirá llevar a cabo ninguna modificación será el identificador de usuario.
	modificarUsuario	**Entrada:** UsuarioForm BindingResult **Salida:** String	Método utilizado para llevar a cabo la modificación de los datos del usuario sobre la base de datos. El resultado será una cadena de caracteres que identificará la pantalla que indica que la operación ha sido realizada correctamente.

Tabla 35.Métodos asociados a la opción "Modificación de usuarios" en la clase UsuarioController.

A continuación se muestran los diagramas de secuencia utilizados para visualizar y documentar la opción "Modificación de usuarios". En el primer diagrama se puede ver la invocación al método obtenerListaUsuarios cuyo objetivo será la visualización final de todos los usuarios dados de alta en la aplicación. Un elemento importante que aparece al principio de este diagrama de secuencia es la utilización de un objeto perteneciente a la clase Hashmap. Este objeto se utilizará para capturar el valor de un usuario seleccionado en esta pantalla, para proceder a su modificación:

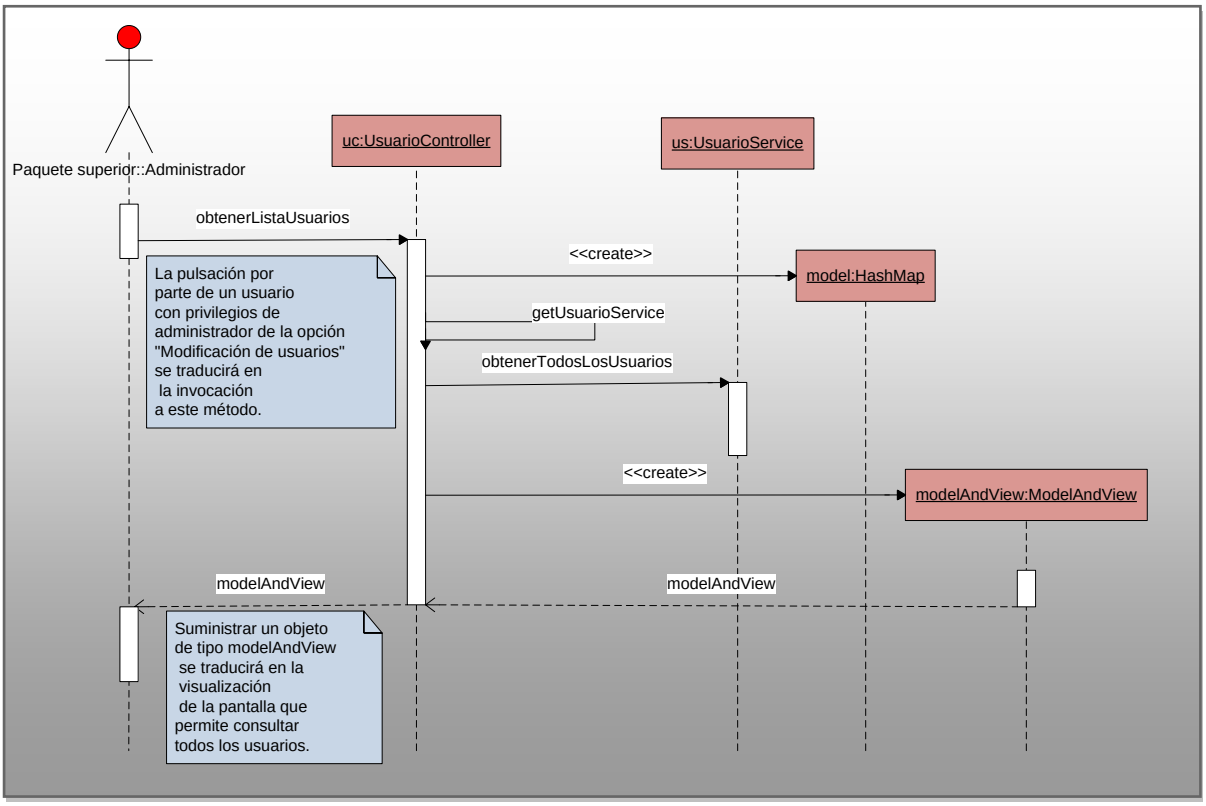


Figura 58. Diagrama de secuencia asociado a la pantalla que permite la visualización de todos los usuarios para permitir su modificación.

En el diagrama de secuencia que aparece en la siguiente figura se representa un escenario en el que un usuario con privilegios de "Administrador" ha seleccionado un usuario para su modificación. En el diagrama se puede observar cómo, tras realizar la consulta del usuario previamente seleccionado, se procede a construir la lista de perfiles que se podrán seleccionar. El resultado final de la interacción será la visualización de un formulario en el que se permitirá modificar los datos de un usuario:

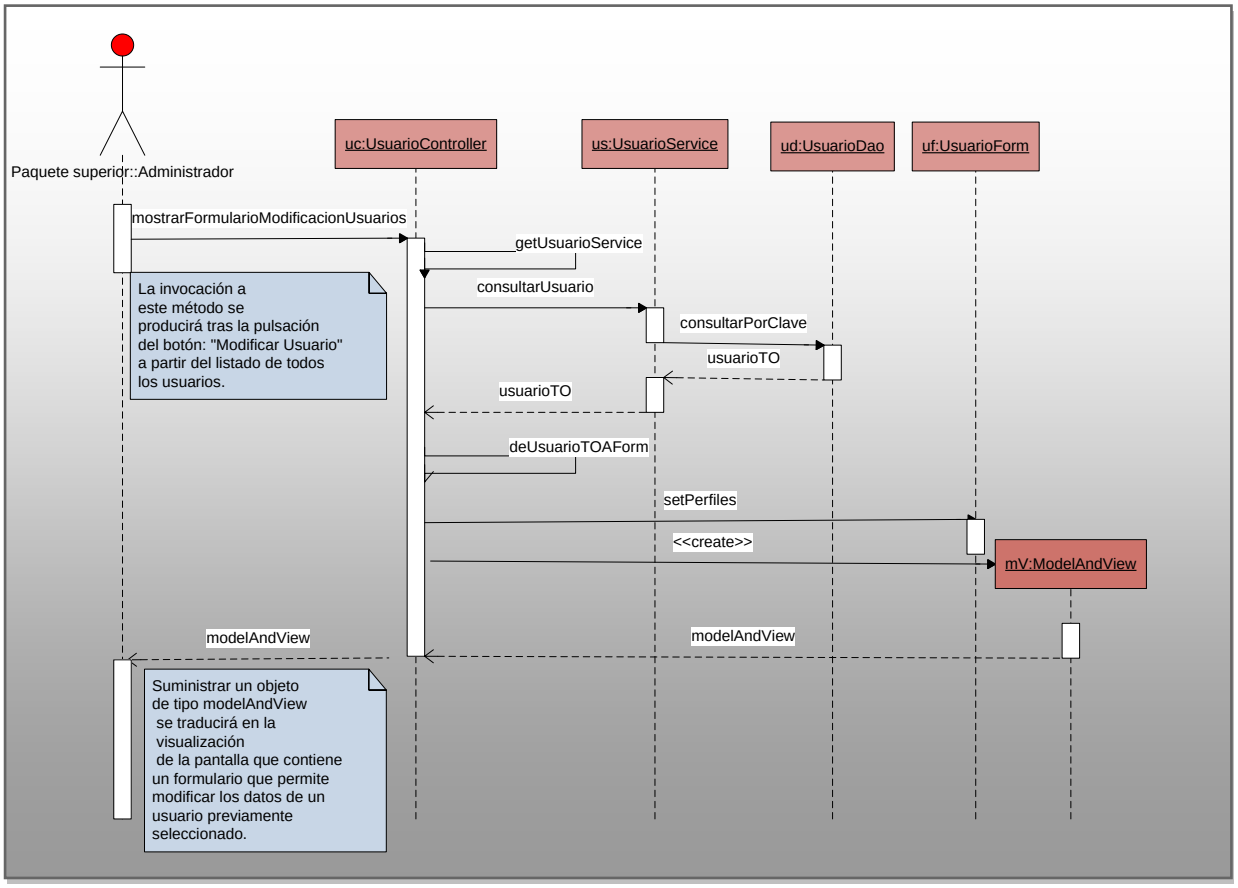


Figura 59. Diagrama de secuencia asociado a la pantalla que permite la visualización del formulario que permite la modificación de un usuario.

Una vez llevadas a cabo las modificaciones sobre el usuario seleccionado, se procederá al almacenamiento de sus datos. El siguiente diagrama de secuencia muestra el escenario en el que un usuario, pulsa el botón "Modificar usuario", tras llevar a cabo las modificaciones:

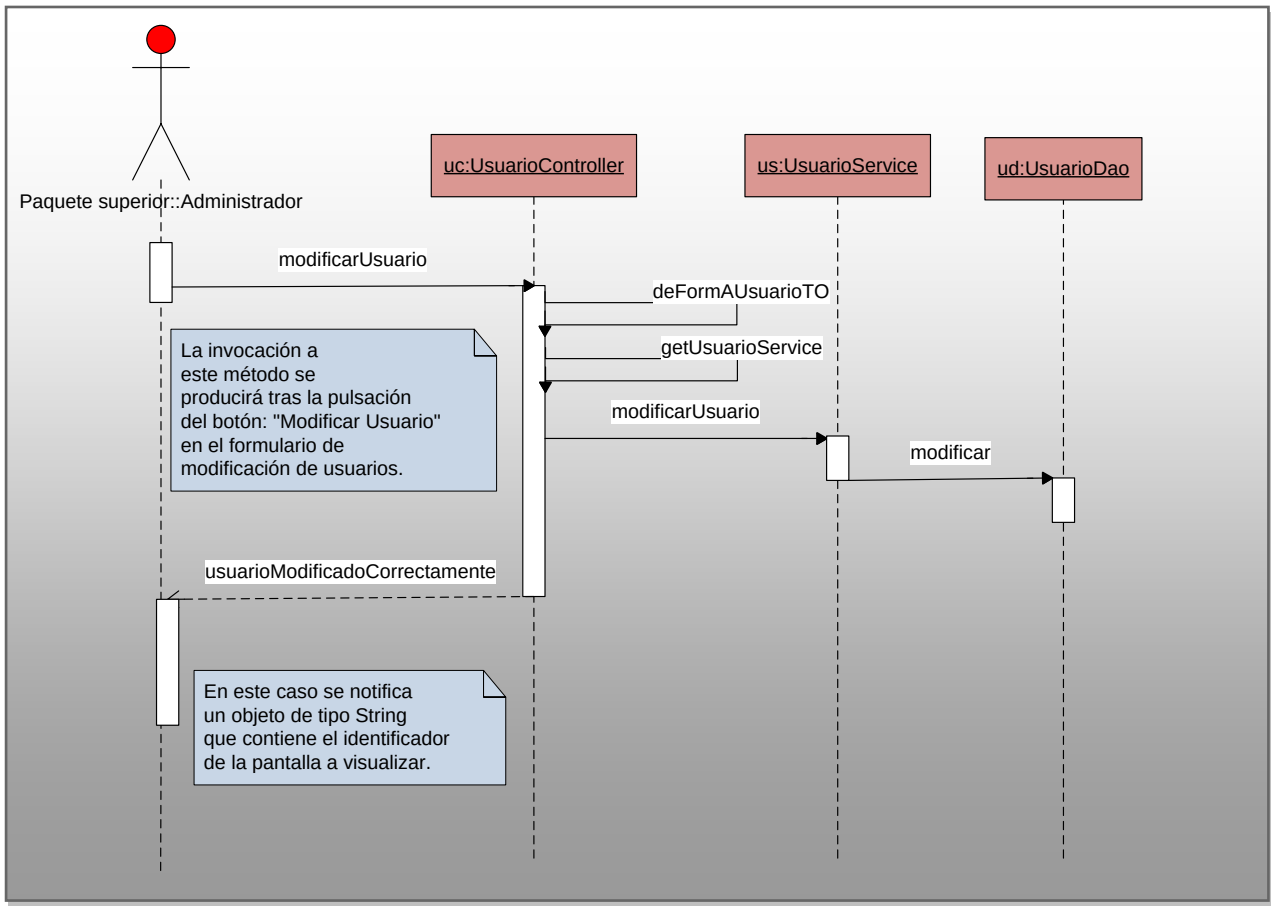


Figura 60. Diagrama de secuencia asociado a la acción de "Modificar usuario".

En el diagrama de secuencia anterior se describe un escenario en el que un usuario pulsa el botón "Modificar usuario" tras llevar a cabo la modificación de algún dato relativo al mismo.

Los ficheros JSP involucrados en la funcionalidad "Modificación de usuarios" son:

- Index.jsp fichero utilizado para presentar la cabecera principal de la aplicación AAS11. Esta cabecera incluye los datos del usuario que se ha validado tras realizar la acción, los botones de ayuda y logout y el menú principal de la aplicación que incluye la totalidad de opciones disponibles.

- IndexSinMenu.jsp: fichero utilizado para presentar la cabecera de la aplicación en pantallas en las que no se pondrán a disposición de los usuarios las opciones de menú de la aplicación (pantallas en las que se visualiza un mensaje informativo como resultado de una acción llevada a cabo por el usuario).

- listaUsuarios.jsp: fichero que permite mostrar un listado de la totalidad de usuarios almacenados en la aplicación.

- modificaciónUsuarios.jsp: fichero que permite presentar un formulario en el que se podrán realizar las modificaciones asociadas a un determinado usuario.

- usuarioModificadoCorrectamente.jsp: fichero que permite la visualización de un mensaje informativo tras la realización de la acción de modificación.

La respuesta HTML que se suministra al navegador tras procesar estos ficheros JSP en el servidor de aplicaciones se muestra en las siguientes figuras:

Figura 61.Pantalla de visualización de la lista de usuarios susceptibles de modificar.

En la figura anterior se puede observar el listado de la totalidad de usuarios susceptibles de modificar. Adicionalmente, se puede ver que la opción de paginación no está implementada, lo que implica que la totalidad de usuarios se muestran en una única página.

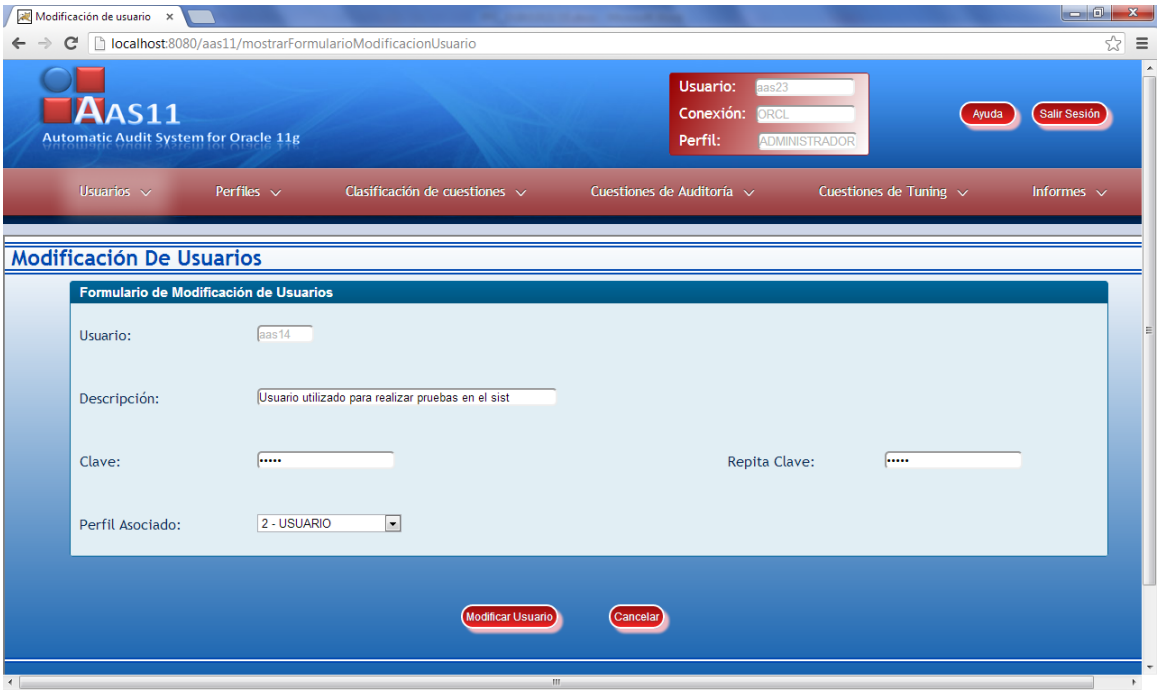


Figura 62.Formulario de modificación de usuarios.

En la figura 62 puede verse el formulario que permite llevar a cabo la modificación de un determinado usuario. Se puede apreciar que, puesto que se trata de un formulario de modificación, no se permite alterar el valor del identificador de usuario.

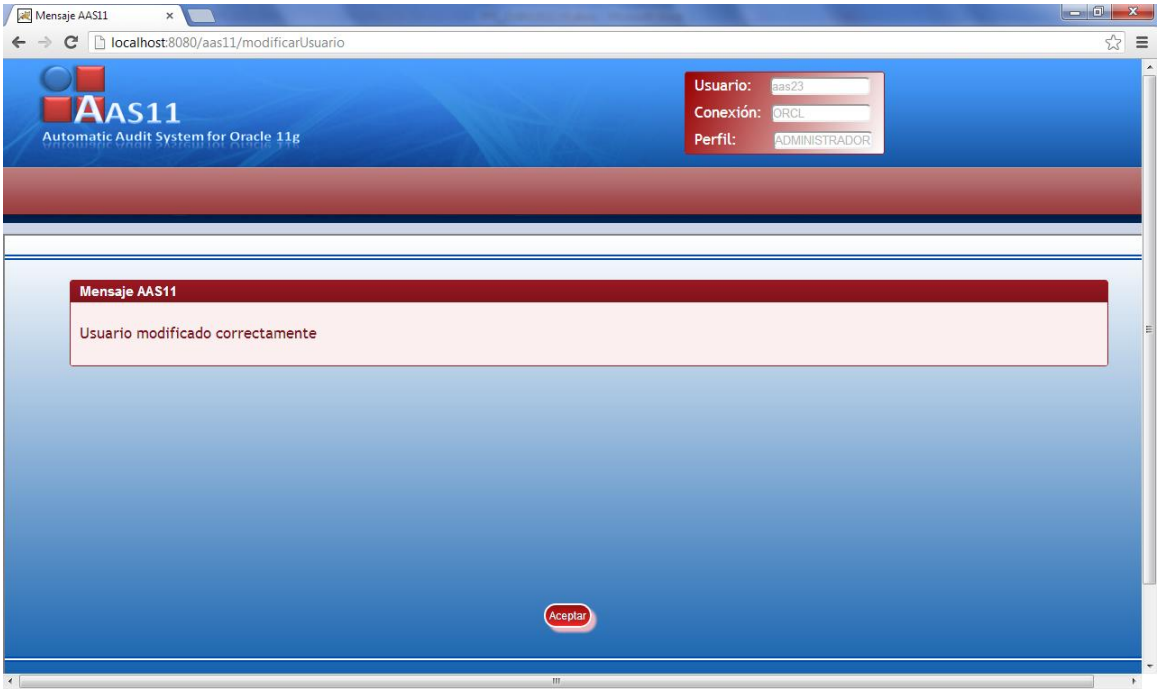


Figura 63. Presentación de mensaje tras la realización de una modificación.

En la figura 63 se visualiza la pantalla en la que se muestra un mensaje informativo tras la realización de una modificación de un usuario.

6.3.3.10 Prueba de opción de Modificación de usuarios

A continuación se muestran cada uno de los casos de prueba aplicados sobre la opción "Modificación de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: selección de la opción "Modificación de usuarios".	
Descripción: después de llevar a cabo la selección de la opción "Modificación de usuarios" se debe mostrar una pantalla que visualizará un listado con la totalidad de usuarios para permitir su selección.	
Fecha de Ejecución: 29-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 36.Prueba unitaria selección de la opción "Modificación de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: visualización del formulario "Modificación de usuarios".	
Descripción: La visualización del formulario "Modificar Usuario" debe incluir los campos: usuario, descripción, clave, repita clave y perfil asociado. Estos campos deben tener la dimensión correcta y en el caso del campo perfil asociado mostrar un desplegable con las opciones posibles. Adicionalmente, el campo usuario, puesto que constituye el identificador del mismo, no debe ser modificable y debe mostrarse en otro color.	
Fecha de Ejecución: 29-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 37.Prueba unitaria visualización del formulario "Modificación de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: actualización correcta utilizando formulario "Modificación de usuarios".	
Descripción: se modificarán varios campos sobre el formulario "Modificación de usuarios". Tras la pulsación del botón "Modificar usuario" se visualizará una pantalla indicando que la modificación ha sido correcta.	
Fecha de Ejecución: 29-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja.
	Acciones sugeridas:

Tabla 38.Prueba unitaria modificación correcta utilizando formulario "Modificación de usuarios".

6.3.3.11 Construcción de opción de Baja de usuarios

Al igual que en el desarrollo de la opción "Modificación de usuarios" se ha utilizado la infraestructura ya desarrollada para llevar a cabo la gestión integral de usuarios. La opción de "Baja de usuarios" se apoyará también en esta infraestructura, de tal forma que la implementación de esta opción únicamente requerirá el desarrollo de una serie de métodos sobre la clase UsuarioController. En la siguiente figura se muestra la clase UsuarioController con los nuevos métodos añadidos:

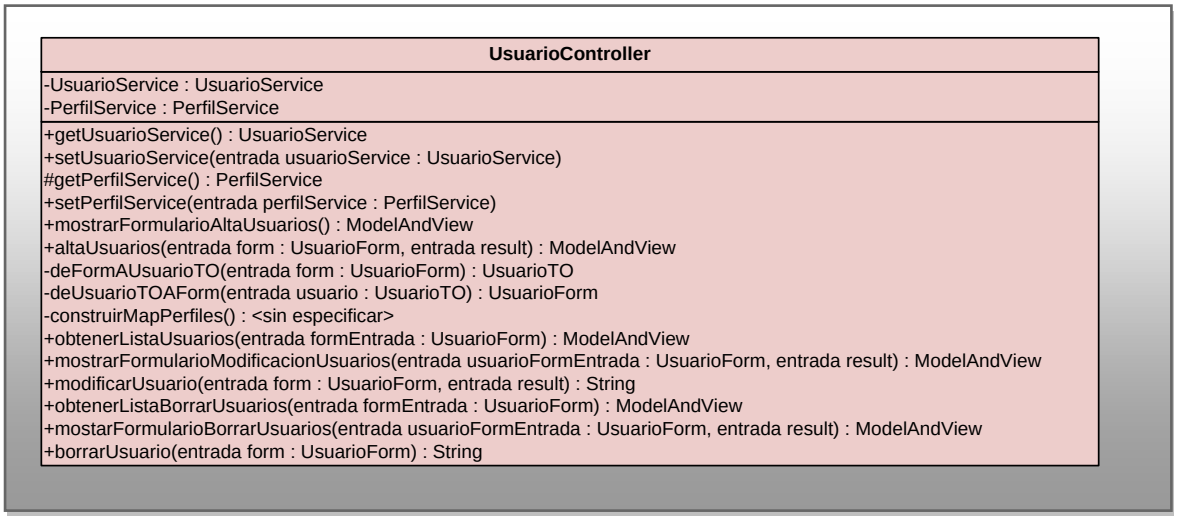


Figura 64. Clase UsuarioController con métodos asociados a la opción "Baja de usuarios".

Los métodos asociados a la opción "Baja de usuarios":

- obtenerListaBorrarUsuarios.
- mostrarFormularioBorrarUsuarios.
- borrarUsuario.

En la siguiente tabla se realiza una descripción de los métodos añadidos, asociados a la funcionalidad "Baja de usuarios":

Clase	Método	Parámetros de entrada y salida	Descripción
UsuarioController	obtenerListaBorrarUsuarios	**Entrada:** UsuarioForm **Salida:** ModelAndView	Método que permite obtener un listado con la totalidad de usuarios almacenados. A partir de este listado el usuario podrá seleccionar un usuario para proceder a su eliminación.
	mostrarFormularioBorrarUsuarios	**Entrada:** UsuarioForm BindingResult **Salida:** No	Método que permite mostrar un formulario con la totalidad de los datos asociados a un usuario para permitir su eliminación.
	borrarUsuario	**Entrada:** UsuarioForm BindingResult **Salida:** String	Método utilizado para llevar a cabo la eliminación del usuario seleccionado. El resultado será una cadena de caracteres que identificará la pantalla que indica que la operación ha sido realizada correctamente.

Tabla 39.Métodos asociados a la opción "Baja de usuarios" en la clase UsuarioController.

A continuación se muestran los diagramas de secuencia utilizados para construir, visualizar y documentar la opción "Baja de usuarios". En el primer diagrama se puede ver la invocación al método obtenerListaBorrarUsuarios cuyo objetivo será la visualización final de todos los usuarios dados de alta en la aplicación. Un elemento importante que aparece al principio de este diagrama de secuencia es la utilización de un objeto perteneciente a la clase Hashmap. Este objeto se utilizará para capturar el valor de un usuario seleccionado en esta pantalla, para proceder a su eliminación[96]:

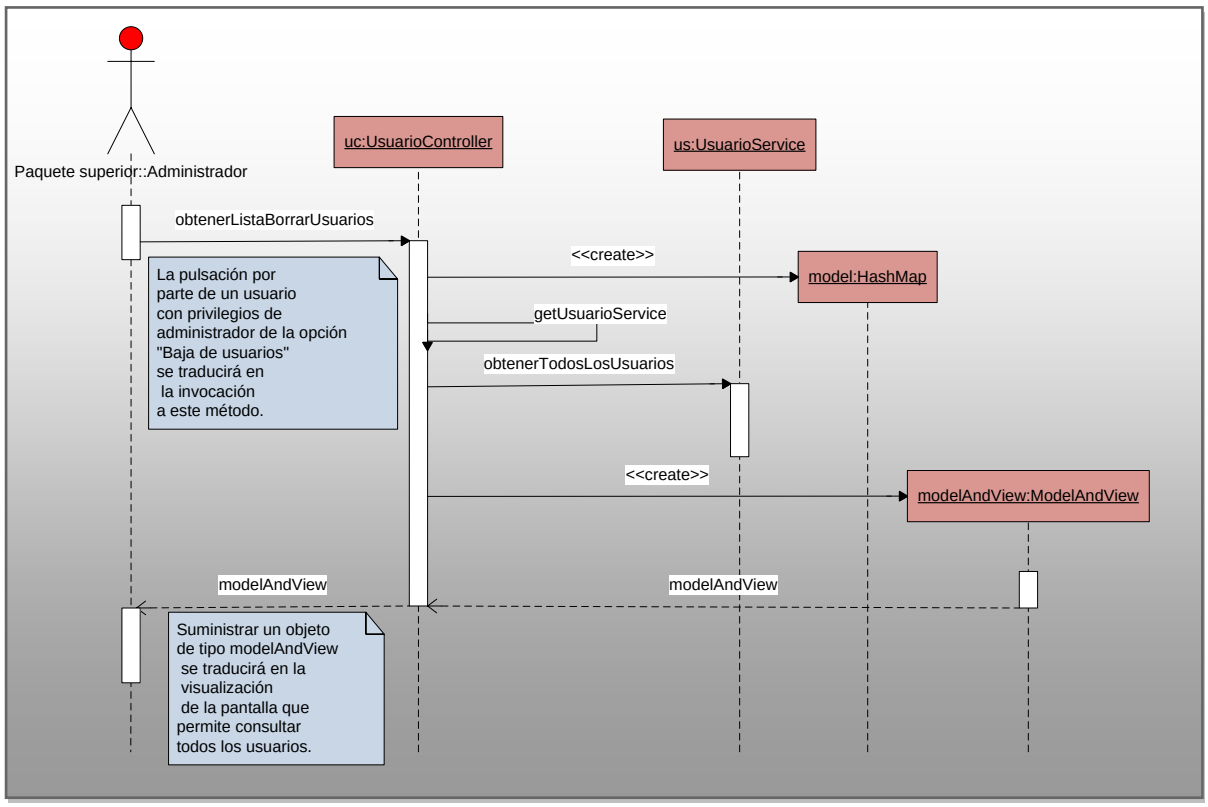


Figura 65. Diagrama de secuencia asociado a la pantalla que permite la visualización de todos los usuarios para permitir su eliminación.

[96] Este diagrama de secuencia es prácticamente idéntico al utilizado en la opción "Modificación de usuarios" puesto que esta opción se apoya en los mismos servicios.

En el diagrama de secuencia que aparece en la siguiente figura se representa un escenario en el que un usuario con privilegios de "Administrador" ha seleccionado un usuario para su eliminación. El resultado final de la interacción será mostrar un formulario en el que se visualizarán los datos de un usuario sin posibilidad de modificación:

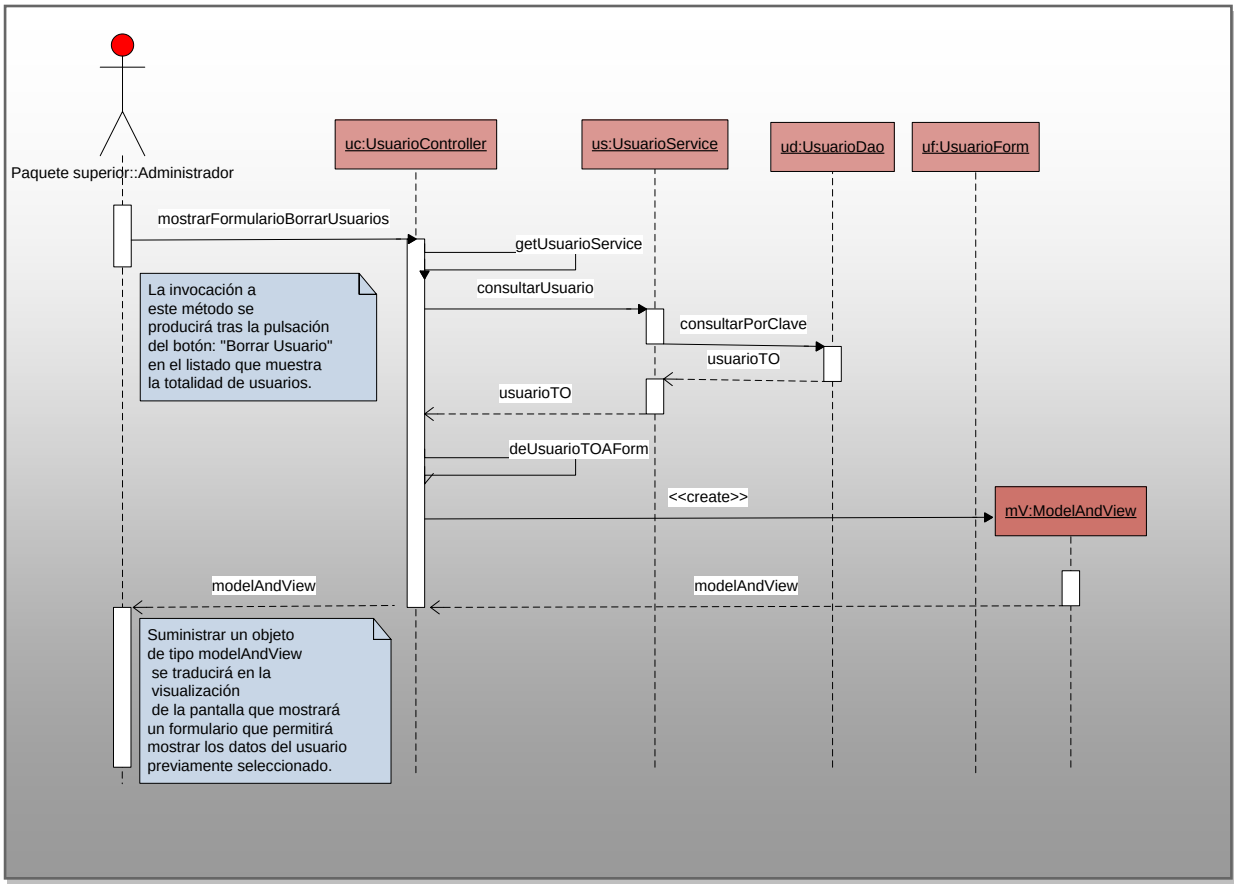


Figura 66. Diagrama de secuencia asociado a la pantalla que permite la visualización del formulario que muestra los datos de un usuario.

Después de verificar que el usuario seleccionado es el que se desea eliminar, se procederá al almacenamiento de sus datos. El siguiente diagrama de secuencia muestra el escenario en el que un usuario, pulsa el botón “Borrar usuario”:

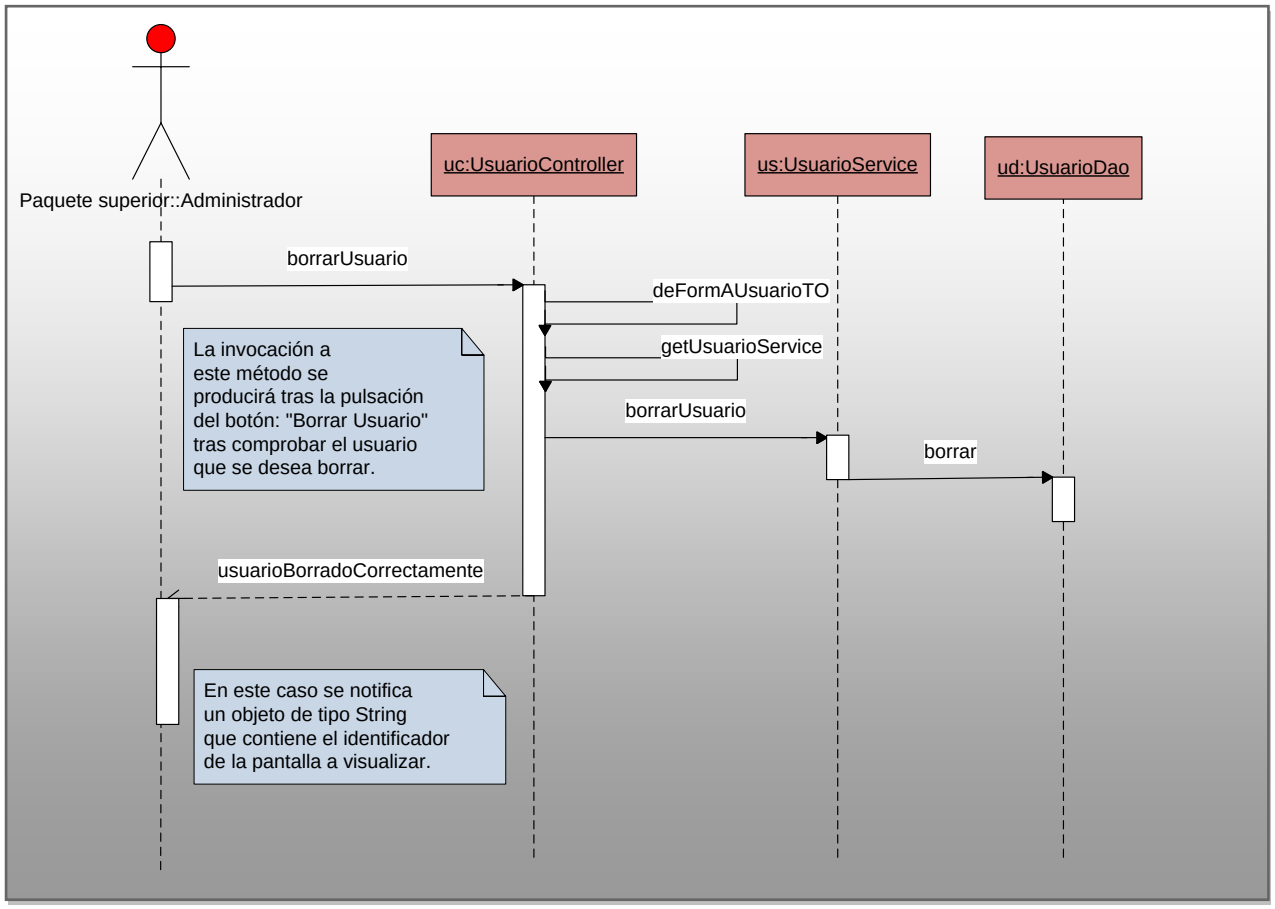


Figura 67. Diagrama de secuencia asociado a la acción de “Borrar usuario”.

En el diagrama de secuencia anterior se describe un escenario en el que un usuario pulsa el botón “Borrar usuario” tras verificar la identidad del usuario que desea borrar.

Los ficheros JSP involucrados en la funcionalidad “Baja de usuarios” son:

- Index.jsp fichero utilizado para presentar la cabecera principal de la aplicación AAS11. Esta cabecera incluye los datos del usuario que se ha validado tras realizar la acción, los botones de ayuda y logout y el menú principal de la aplicación que incluye la totalidad de opciones disponibles.

- IndexSinMenu.jsp: fichero utilizado para presentar la cabecera de la aplicación en pantallas en las que no se pondrán a disposición de los usuarios las opciones de menú de la aplicación (pantallas en las que se visualiza un mensaje informativo como resultado de una acción llevada a cabo por el usuario).

- listaBorrarUsuarios.jsp: fichero que permite mostrar un listado de la totalidad de usuarios almacenados en la aplicación.

- borrarUsuarios.jsp: fichero que permite presentar un formulario en el que se podrán visualizar los datos del usuario a eliminar.

- usuarioBorradoCorrectamente.jsp: fichero que permite la visualización de un mensaje informativo tras la realización de la acción de borrado.

La respuesta HTML que se suministra al navegador tras procesar estos ficheros JSP en el servidor de aplicaciones se muestra en las siguientes figuras:

Figura 68.Pantalla de visualización de la lista de usuarios susceptibles de borrado.

En la figura anterior se puede observar el listado de la totalidad de usuarios susceptibles de borrarse. Adicionalmente, se puede ver que la opción de paginación no está implementada, lo que implica que la totalidad de usuarios se muestran en una única página.

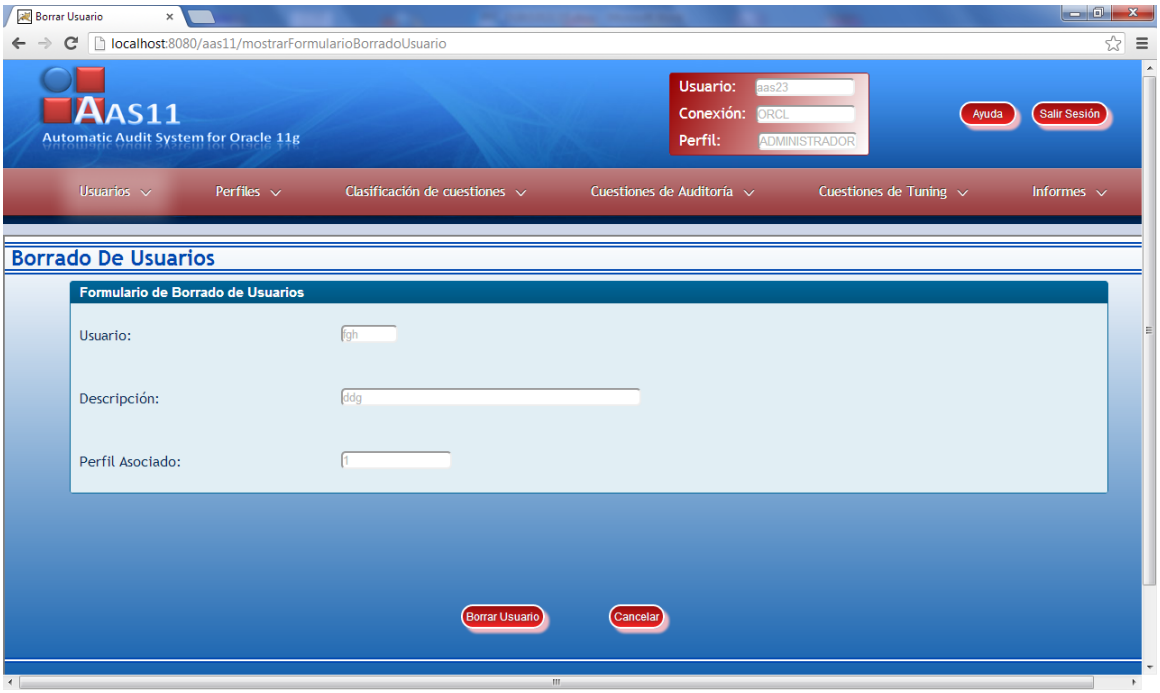


Figura 69.Formulario de borrado de usuarios.

En la figura 69 puede verse el formulario que permite llevar a cabo la eliminación de un usuario seleccionado con anterioridad. Se puede apreciar que, puesto que se trata de un formulario de borrado, los valores de los campos se muestran bloqueados.

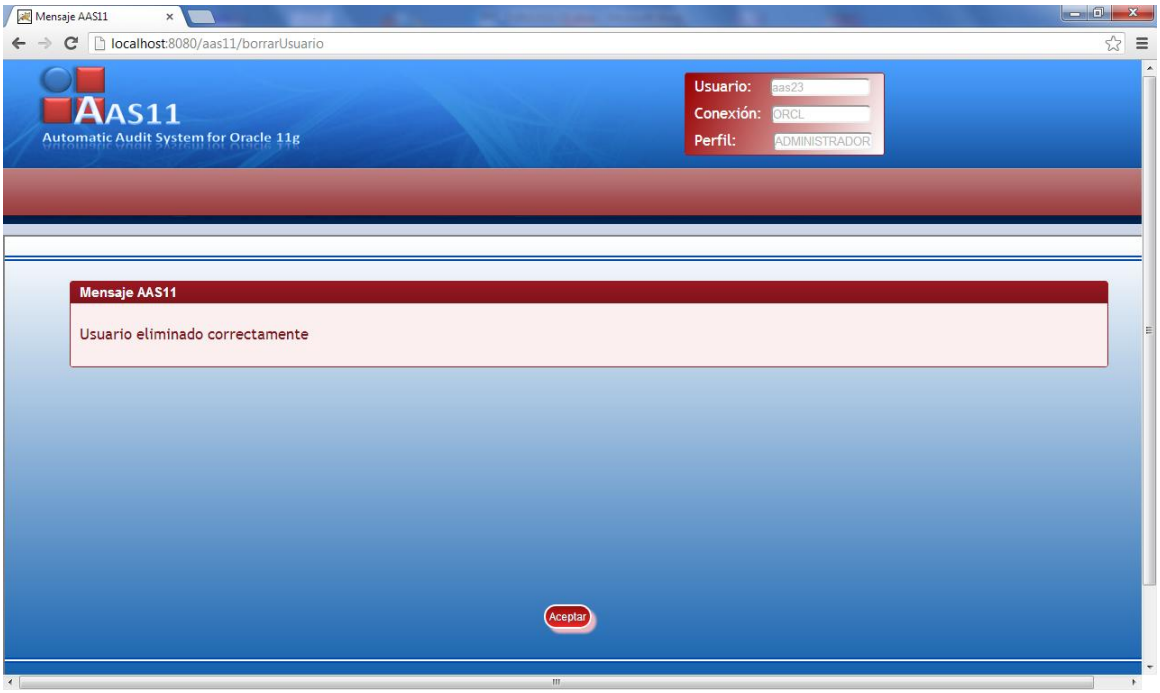


Figura 70. Presentación de mensaje tras la realización de un borrado.

En la figura 70 se visualiza la pantalla en la que se muestra un mensaje informativo tras la realización de una acción de borrado.

6.3.3.12 Prueba de opción de Baja de usuarios

A continuación se muestran cada uno de los casos de prueba aplicados sobre la opción "Borrado de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: selección de la opción "Baja de usuarios".	
Descripción: después de llevar a cabo la selección de la opción "Baja de usuarios" se debe mostrar una pantalla que visualizará un listado con la totalidad de usuarios para permitir su selección.	
Fecha de Ejecución: 02-11-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 40.Prueba unitaria selección de la opción "Baja de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: visualización del formulario "Borrado de usuarios".	
Descripción: la visualización del formulario "Borrado de usuarios" debe incluir los campos: usuario, descripción y perfil asociado. Estos campos deben tener la dimensión y deben ser visualizados para impedir su modificación.	
Fecha de Ejecución: 02-11-2012.	
Responsable: equipo de desarrollo.	
Resultado: fallo.	
Defectos detectados	
Defecto 1	**Resumen:** En este formulario se detecta un problema de visualización asociado al campo perfil. En lugar de visualizar el perfil junto con su descripción, se visualiza únicamente el identificador del perfil.
	Prioridad: media
	Acciones sugeridas: Revisión del método mostrarFormularioBorrarUsuarios perteneciente a la clase UsuarioController. Sobre este método se deben buscar problemas asociados a la construcción del mapeo de perfiles. El fallo es detectado concretamente en este punto y se procede a su subsanación.

Tabla 41.Prueba unitaria visualización del formulario "Borrado de usuarios".

Datos de la prueba	
Nombre del Caso de Prueba: actualización correcta utilizando formulario "Modificación de usuarios".	
Descripción: se modificarán varios campos sobre el formulario "Modificación de usuarios". Tras la pulsación del botón "Modificar usuario" se visualizará una pantalla indicando que la modificación ha sido correcta.	
Fecha de Ejecución: 29-10-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja.
	Acciones sugeridas:

Tabla 42.Prueba unitaria eliminación correcta utilizando formulario "Borrado de usuarios".

6.3.4 Tareas asociadas al requisito PB-0-003

6.3.4.1 Diagrama de casos de uso

En el siguiente diagrama de casos de uso aparece la representación de la gestión de perfiles llevada a cabo por la aplicación. Inicialmente, únicamente se permitirá la consulta de los perfiles utilizados en la aplicación a un usuario con privilegios de "Administrador". Esta consulta deberá detallar tanto los perfiles existentes, como su relación con las opciones asociadas y con los menús de los que dependen estas opciones.

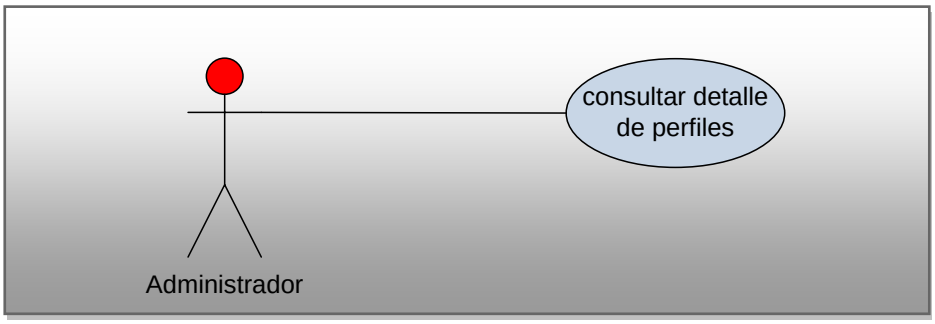


Figura 71.Diagrama de casos de uso "Perfiles de usuario".

6.3.4.2 Incorporación en el modelo de datos

Como se puede observar en el apartado "6.3.3.3 Elaboración del modelo de datos" tanto los tipos de entidad consideradas en el modelo conceptual, como las tablas asociadas a estos tipo de entidad son las necesarias para contemplar el requisito PB-0-003.

Los requisitos PB-0-002 y PB-0-003 están estrechamente relacionados y el motivo por el que se ha tomado la decisión de incluir en el modelo de datos estas tablas lo antes posible reside en la necesidad de llevar a cabo pruebas "reales" sobre la funcionalidad asociada a la gestión de usuarios.

6.3.4.3 Carga de datos en la base de datos

En este caso el procedimiento de carga únicamente incluiría aquellos elementos asociados a la incorporación de esta opción de menú en pantalla: Alta de opciones y menús utilizados para la visualización de la opción de "Perfiles de usuario" en la pantalla. Cada una de estas tablas se utilizará con la siguiente finalidad:

- Menús: almacenará los identificadores de cada uno de los menús utilizados, el título de cada uno de los menús (dato que se utiliza para visualizar cada uno de ellos) y una breve descripción sobre su contenido y propósito.

- Opciones: almacenará el detalle de cada una de las opciones disponibles para cada uno de los menús. Este detalle incluye un identificador de la opción, el título (que es utilizado para visualizar la opción), el identificador del menú al que está asociada la opción y el identificador del perfil que puede utilizar dicha opción.

6.3.4.4 Construcción de opción de Consulta de detalles de perfiles

El diagrama de clases asociado a la funcionalidad de "Consulta de perfiles/menús/opciones" se muestra en la siguiente figura:

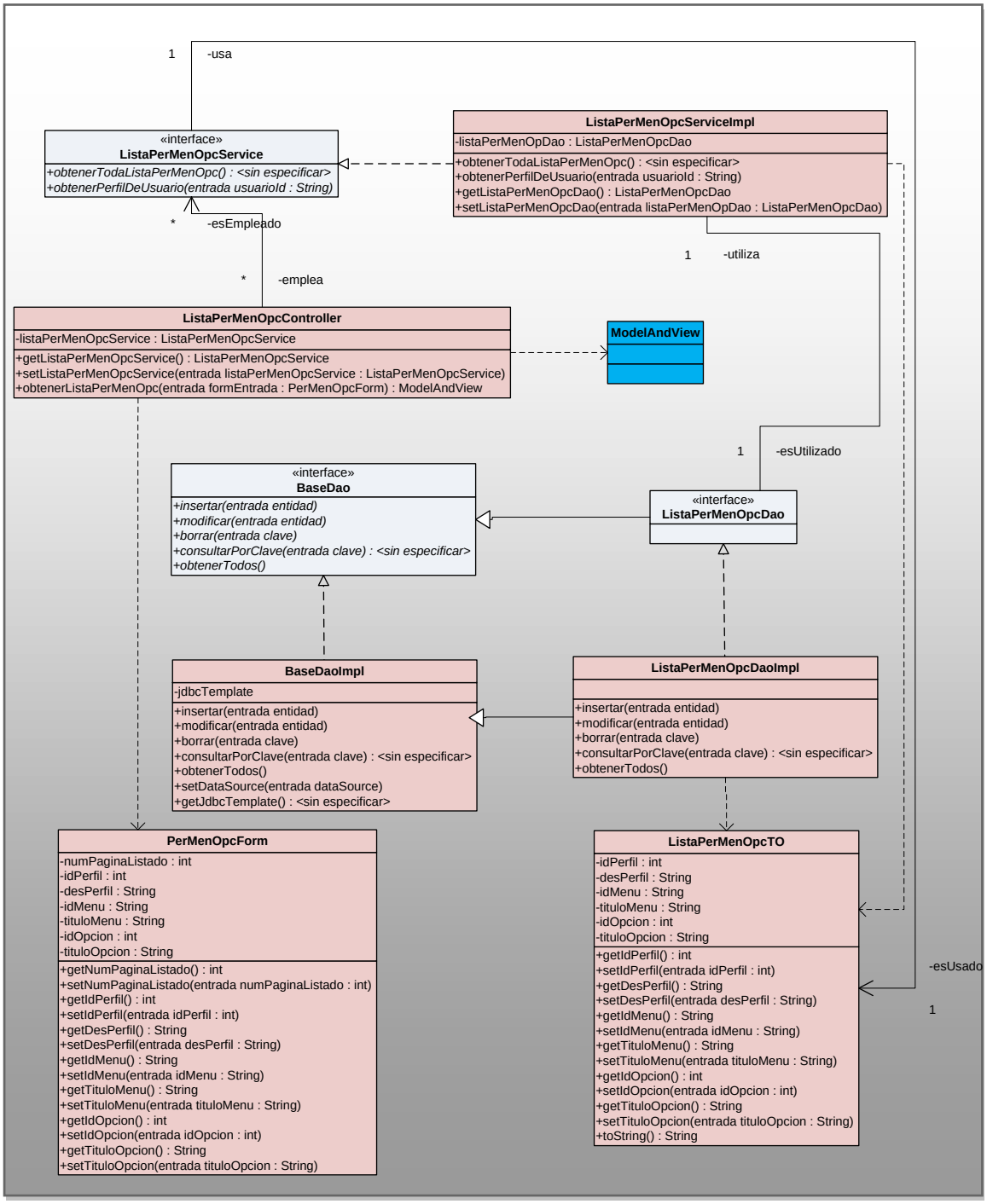


Figura 72. Diagrama de clases asociado a la visualización de "Perfiles/Menús/Opciones".

En el diagrama de clases representado en la figura 72, se puede observar que se utilizan los mismos patrones de diseño que en el diagrama que representa la gestión de usuarios. El color de la clase ModelAndView indica que se trata de una clase que no ha sido implementada por el desarrollador. A la hora de interpretar el diagrama de clases se expondrá el orden de implementación de las mismas junto con una breve explicación del cometido de cada una de ellas:

1. ListaPerMenOpcTO: clase utilizada para el intercambio de información con la base de datos. Los atributos que se definen en esta clase representan el conjunto de campos que se visualizarán asociados al listado de perfiles, menús y opciones.

2. ListaPerMenOpcDaoImpl: clase en la que se implementan los métodos asociados a las acciones de consulta sobre la base de datos. La totalidad de clases DAO implementadas extenderán la clase BaseDao, en la que se definen los métodos que permiten la obtención y manipulación de los conjuntos de datos procedentes de la base de datos. Tanto sobre la clase BaseDao, como sobre la clase ListaPerMenOpcDaoImpl se definen interfaces. La interfaz asociada a la clase ListaPerMenOpcDaoImpl extiende la interfaz BaseDao.

3. ListaPerMenOpcServiceImpl: clase utilizada para definir los servicios involucrados en la visualización del listado que relaciona los perfiles, menús y opciones utilizados en la aplicación AAS11. Sobre esta clase se define una interfaz que será utilizada por la clase conroller correspondiente. El servicio que suministra esta clase es la obtención de los datos a visualizar en el listado de perfiles, menús y opciones.

4. PerMenOpcForm: clase que representa el conjunto de información intercambiada en el formulario utilizado en la visualización de perfiles, menús y opciones.

5. ListaPerMenOpcController: esta clase se utilizará como elemento coordinador, invocando las acciones correspondientes para llevar a cabo la visualización del listado que contiene las relaciones entre perfiles, menús y opciones.

En las siguientes tablas aparece una breve descripción de la funcionalidad implementada en cada uno de los métodos pertenecientes a estas clases:

Clase	Método	Parámetros de entrada y salida	Descripción
ListaPerMenOpcTO	getIdPerfil	**Entrada:** No **Salida:** int	Método que permite la obtención del identificador de perfil.
	setIdPerfil	**Entrada:** int **Salida:** No	Método que permite establecer el valor del identificador de perfil.
	getDesPerfil	**Entrada:** No **Salida:** String	Método que permite la obtención de la descripción de un perfil.
	settDesPerfil	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la descripción de un perfil.
	getIdMenu	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de menú.
	setIdMenu	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de menú.
	getTituloMenu	**Entrada:** No **Salida:** String	Método que permite la obtención del título del menú.
	setTituloMenu	**Entrada:** String **Salida:** No	Método que permite establecer el valor del título del menú.
	getIdOpcion	**Entrada:** No **Salida:** int	Método que permite la obtención del identificador de opción.
	setIdOpcion	**Entrada:** int **Salida:** No	Método que permite establecer el valor del identificador de la opción.
	getTituloOpcion	**Entrada:** String **Salida:** No	Método que permite establecer el valor del título de la opción.
	setTituloOpcion	**Entrada:** String **Salida:** No	Método que permite establecer el valor del título de la opción.

Tabla 43.Tabla de descripción de métodos asociados a las clase ListaPerMenOpcTO.

Clase	Método	Parámetros de entrada y salida	Descripción
ListaPerMenOpcDaoImpl	insertar	**Entrada:** ListaPerMenOpcTO **Salida:** No	No utilizado. Sin implementar.
	modificar	**Entrada:** ListaPerMenOpcTO **Salida:** String	No utilizado. Sin implementar.
	borrar	**Entrada:** String **Salida:** No	No utilizado. Sin implementar.
	consultarPorClave	**Entrada:** String **Salida:** UsuarioTO	No utilizado. Sin implementar.
	obtenerTodos	**Entrada:** No **Salida:** List<UsuarioTO>	Método que permite obtener la totalidad de usuarios almacenados en la base de datos.
	obtenerPerfilDeUsuario	**Entrada:** String **Salida:** ListaPerMenOpcTO	Método utilizado para obtener el perfil asignado a un determinado usuario.

Tabla 44.Tabla de descripción de métodos asociados a las clase ListaPerMenOpcDaoImpl.

Clase	Método	Parámetros de entrada y salida	Descripción
ListaPerMenOpcServiceImpl	getListaPerMenOpcDao	**Entrada:** No **Salida:** ListaPerMenOpcDao	Método que permite obtener el valor del atributo ListaPerMenOpc-Dao.
	setListaPerMenOpcDao	**Entrada:** ListaPerMenOpcDao **Salida:** No	Método que permite establecer el valor del atributo ListaPerMenOpc-Dao.
	obtenerTodaListaPerMenOpc	**Entrada:** No **Salida:** List<ListaPerMenOpcTO>	Método que permite obtener una lista que contendrá el detalle de los datos asociados a la relaciones entre perfiles, menús y opciones.

Tabla 45.Tabla de descripción de métodos asociados a la clase ListaPerMenOpcServiceImpl.

Clase	Método	Parámetros de entrada y salida	Descripción
PerMenOpcForm	getNumPaginaListado	**Entrada:** No **Salida:** int	Método que permite obtener el valor del atributo numPaginaListado que será utilizado para implementar la funcionalidad de paginación.
	setNumPaginaListado	**Entrada:** int **Salida:** No	Método que permite establecer el valor del atributo numPaginaListado que será utilizado para implementar la funcionalidad de paginación.
	getIdPerfil	**Entrada:** No **Salida:** int	Método que permite la obtención del identificador de perfil.
	setIdPerfil	**Entrada:** int **Salida:** No	Método que permite establecer el valor del identificador de perfil.
	getDesPerfil	**Entrada:** No **Salida:** String	Método que permite la obtención de la descripción de perfil.
	setDesPerfil	**Entrada:** String **Salida:** No	Método que permite establecer el valor de la descripción de perfil.
	getIdMenu	**Entrada:** No **Salida:** String	Método que permite la obtención del identificador de menú.
	setIdMenu	**Entrada:** String **Salida:** No	Método que permite establecer el valor del identificador de menú.
	getTituloMenu	**Entrada:** No **Salida:** String	Método que permite la obtención del título del menú.
	setTituloMenu	**Entrada:** String **Salida:** No	Método que permite establecer el valor del título del menú.
	getIdOpcion	**Entrada:** No **Salida:** int	Método que permite la obtención del identificador de opción.
	setIdOpcion	**Entrada:** int **Salida:** No	Método que permite establecer el valor del identificador de opción.
	getTituloOpcion	**Entrada:** No **Salida:** String	Método que permite la obtención del título de la opción.
	setTituloOpcion	**Entrada:** No **Salida:** String	Método que permite establecer el título de la opción.

Tabla 46.Tabla de descripción de métodos asociados a la clase PerMenOpcForm.

Clase	Método	Parámetros de entrada y salida	Descripción
ListaPerMenOpcController	getListaPerMenOpcService	**Entrada:** No **Salida:** ListaPerMenOpcService	Método que permite obtener el valor del atributo listaPerMenOpcService, utilizado para acceder a las funcionalidades implementadas por la clase ListaPerMenOpcService.
	setListaPerMenOpcService	**Entrada:** ListaPerMenOpcService **Salida:** No	Método que permite establecer el valor del atributo listaPerMenOpcService, utilizado para acceder a las funcionalidades implementadas por la clase ListaPerMenOpcService.
	obtenerListaPerMenOpc	**Entrada:** PerMenOpcForm **Salida:** ModelAndView	Método que permite visualizar la lista que representa las relaciones entre perfiles, menús y opciones.

Tabla 47.Tabla de descripción de métodos asociados a la clase ListaPerMenOpcController.

A continuación se muestran los diagramas de secuencia utilizados para construir, visualizar y documentar la opción "Consulta perfiles/menús/opciones". En el primer diagrama se puede ver la invocación al método obtenerListaPerMenOpc cuyo objetivo será presentar un listado que permita presentar las relaciones existentes entre perfiles, menús y opciones.

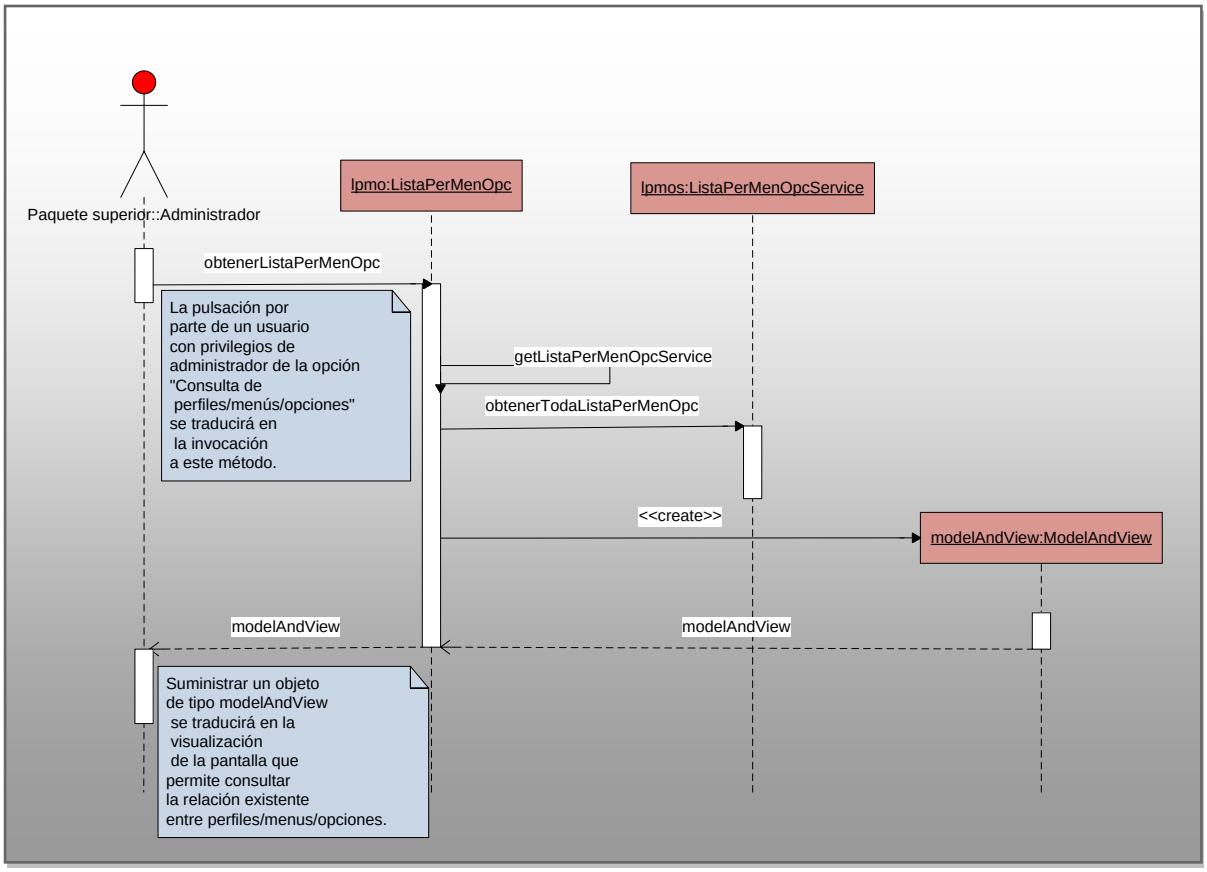


Figura 73. Diagrama de secuencia asociado a la pantalla que permite la visualización de las relaciones existentes entre perfiles, menús y opciones.

Los ficheros JSP involucrados en la funcionalidad "Consulta de perfiles/menús/opciones" son:

- Index.jsp: fichero utilizado para presentar la cabecera principal de la aplicación AAS11. Esta cabecera incluye los datos del usuario que se ha validado tras realizar la acción, los botones de ayuda y logout, y el menú principal de la aplicación que incluye la totalidad de opciones disponibles.

- listaPerMenOpc.jsp: fichero que permite mostrar un listado de las relaciones existentes entre los diferentes perfiles, menús y opciones utilizados en la aplicación.

La respuesta HTML que se suministra al navegador tras procesar estos ficheros JSP en el servidor de aplicaciones se muestra en las siguientes figuras:

Id Perfil	Perfil	Id Menú	Título Menú	Id Opción	Opción
1	ADMINISTRADOR	1	Usuarios	1	Alta de usuarios
1	ADMINISTRADOR	1	Usuarios	2	Modificación de usuarios
1	ADMINISTRADOR	1	Usuarios	3	Baja de usuarios
1	ADMINISTRADOR	1	Usuarios	20	Alta de usuarios
1	ADMINISTRADOR	2	Perfiles	4	Consulta de detalle de perfiles
1	ADMINISTRADOR	3	Clasificación de cuestiones	10	Baja de apartados de cuestiones de tuning
1	ADMINISTRADOR	3	Clasificación de cuestiones	9	Modificación de apartados de cuestiones de tuning
1	ADMINISTRADOR	3	Clasificación de cuestiones	8	Alta de apartados de cuestiones de tuning
1	ADMINISTRADOR	3	Clasificación de cuestiones	7	Baja de secciones de cuestiones de auditoría
1	ADMINISTRADOR	3	Clasificación de cuestiones	5	Alta de secciones de cuestiones de auditoría
1	ADMINISTRADOR	3	Clasificación de cuestiones	6	Modificación de secciones cuestiones de auditoría
1	ADMINISTRADOR	4	Cuestiones de auditoría	11	Alta de cuestiones de auditoría
1	ADMINISTRADOR	4	Cuestiones de auditoría	13	Baja de cuestiones de auditoría
1	ADMINISTRADOR	4	Cuestiones de auditoría	12	Modificación de cuestiones de auditoría

Figura 74.Pantalla de visualización de las relaciones existentes entre perfiles, menús y opciones.

En la figura anterior se puede observar el listado que contempla la totalidad de relaciones existentes entre perfiles, menús y opciones. Adicionalmente, se puede ver que la opción de paginación no está implementada, lo que implica que la totalidad de estas relaciones se muestran en una única página.

6.3.4.5 Prueba de opción de Consulta de detalle de perfiles

A continuación se muestran cada uno de los casos de prueba aplicados sobre la opción "Consulta de perfiles, menús y opciones".

Datos de la prueba	
Nombre del Caso de Prueba: selección de la opción "Consulta de perfiles/menús/opciones".	
Descripción: después de llevar a cabo la selección de la opción "Consulta de perfiles/menús/opciones" se debe mostrar una pantalla que contemplará las relaciones existentes entre perfiles, menús y opciones. Sobre este listado deben aparecer los campos: Id Perfil, Perfil, Id Menú, Título Menú, Id Opción y Opción. Se debe verificar la corrección en su visualización y contenido.	
Fecha de Ejecución: 04-11-2012.	
Responsable: equipo de desarrollo.	
Resultado: acierto.	
Defectos detectados	
Defecto 1	**Resumen:**
	Prioridad: alta/media/baja
	Acciones sugeridas:

Tabla 48.Prueba unitaria selección de la opción "Consulta de Perfiles, menús y opciones".

6.3.5 Reunión de revisión y retrospectiva del Sprint 1

En esta reunión se realiza una revisión de la funcionalidad implementada en el Sprint 1 con la finalidad de llevar a cabo una evaluación de la misma. El resultado final es la obtención de una validación satisfactoria.

Como elementos a destacar, por su particular dificultad, en esta reunión se plantean los problemas relativos a la realización de un prototipo de pantallas con un aspecto prácticamente equivalente al resultado definitivo, en un estadio del desarrollo tan temprano. También se destaca la complejidad del establecimiento del modelo de arquitectura que se utilizará durante todo el proceso de desarrollo.

En la figura que se muestra a continuación se puede ver el diagrama Burn-Down obtenido tras la realización del Sprint 1:

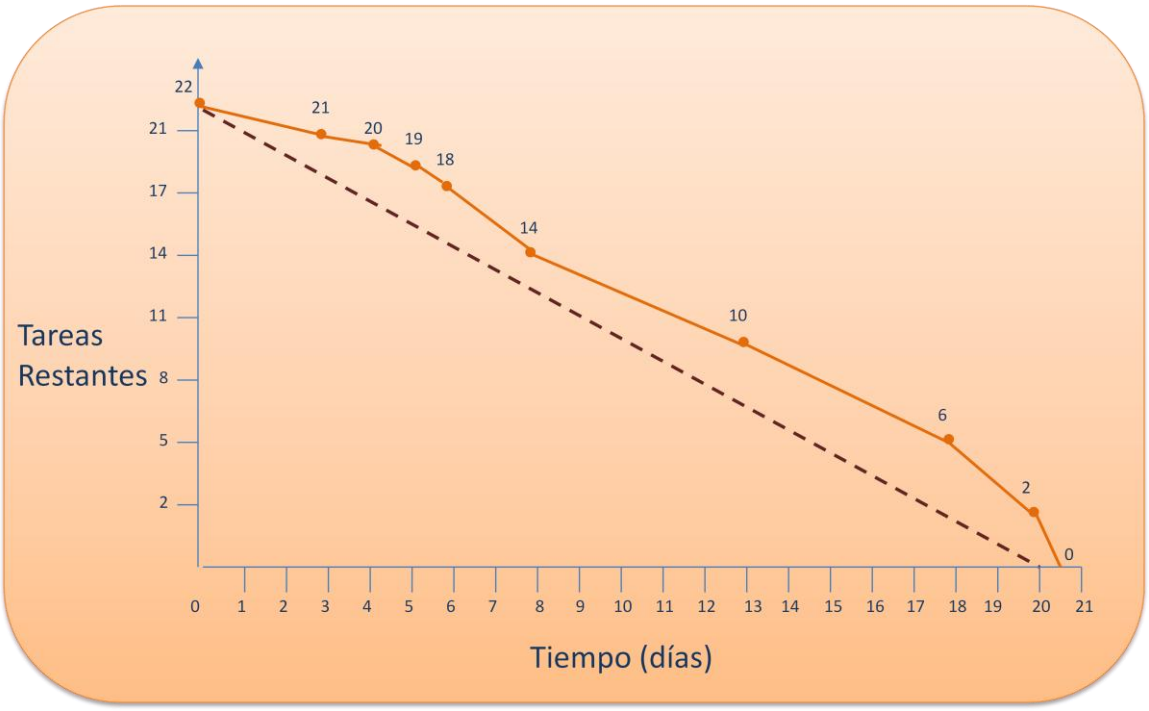


Figura 75.Diagrama Burn-Down asociado al Sprint 1.

En el diagrama Burn-Down de la figura se puede apreciar que la duración inicial estimada del Sprint 1 estaba estimada en cuatro semanas. Finalmente, para llevar a cabo la totalidad de las tareas planificadas se ha invertido medio día más. Los esfuerzos más importantes se han realizado en la selección de la arquitectura, elaboración de un prototipo inicial, en la construcción del procedimiento de login y adicionalmente, en la construcción de las funcionalidades que involucran un mayor número de pantallas de presentación.

En la siguiente tabla se muestra el Sprint Backlog asociado al Sprint 1. En esta tabla se puede apreciar como al requisito PB-0-001 se le asigna un porcentaje completado del 60% puesto que, aunque se ha realizado el esfuerzo principal en este Sprint para su diseño e implementación, podría sufrir modificaciones. El resto de requisitos se consideran prácticamente implementados (90%) a falta de incluir la opción de paginación.

ID	Nombre	Prioridad	Coste	Completado	Sprint 1
PB-0-001	Interfaz web	0.9	70	60%	60%
PB-0-002	Multiusuario	0.9	25	100%	90%
PB-0-003	Perfiles de usuario	0.8	18	100%	90%

Tabla 49.Sprint Backlog 1 tras la consecución del Sprint 1.

6.4 Principios de descripción en el resto de Sprints

El Sprint 1 se ha descrito de forma detallada, destacando el proceso de diseño utilizado para implementar las funcionalidades escogidas en la reunión de planificación de dicho Sprint. Con la finalidad de que la descripción del resto de Sprints aporte información complementaria[97], se ha optado por presentar una síntesis de los Sprints 2, 3 y 4 en la que se destacarán aquellos elementos técnicos considerados como especialmente relevantes en la obtención de soluciones.

6.5 Síntesis del Sprint 2

6.5.1 Reunión de planificación del Sprint 2

A lo largo del Sprint 1 se han establecido los fundamentos arquitectónicos de la aplicación AAS11 y se han comenzado a implementar las primeras funcionalidades. El propósito del Sprint 2 será el desarrollo de parte de las funcionalidades asociadas a la gestión de los datos manejados por la aplicación AAS11. En concreto se escogen los siguientes requisitos:

- PB-0-004 Definición de cuestiones de auditoría: este requisito establece la necesidad de definir cuestiones de auditoría, indicando la información que debe contemplarse asociada a cada una de estas cuestiones.

- PB-0-005 Gestión de cuestiones de auditoría: este requisito está estrechamente relacionado con el requisito PB-0-004 y determina que debe permitirse la gestión de las cuestiones de auditoría implementando las acciones de alta, baja y modificación de cuestiones.

- PB-0-008 Definición de cuestiones de tuning: en este caso la aplicación debe contemplar la posibilidad de definir cuestiones de tuning. El detalle del contenido de una cuestión de tuning aparece perfectamente contemplado en el requisito.

- PB-0-009 Gestión de las cuestiones de tuning: estrechamente asociado al requisito PB-0-008 aparece este requisito, en el que se establece la necesidad de gestionar las cuestiones de tuning permitiendo acciones de alta, baja y modificación.

[97] Incluso en el Sprint 1 ya se pueden observar grandes similitudes entre los diagramas de clases y eventos que describen funcionalidades distintas. Esta es una consecuencia de la aplicación de patrones de diseño, que permiten homogeneizar la obtención de soluciones.

El Sprint Backlog asociado al Sprint 2 se muestra a continuación:

ID	Nombre	Prioridad	Coste	Completado	Sprint 1	Sprint 2
PB-0-004	Definición de cuestiones de auditoría	0.8	30	0	0	0
PB-0-005	Gestión de cuestiones de auditoría	0.7	40	0	0	0
PB-0-008	Definición de cuestiones de tuning	0.8	30	0	0	0
PB-0-009	Gestión de cuestiones de tuning	0.7	40	0	0	0

Tabla 50.Sprint Backlog 2.

Al repasar la definición de estos requisitos podemos identificar elementos comunes en su contenido. Estos elementos comunes se describen en la siguiente tabla:

Contenido del requisito	Cuestiones de auditoría	Cuestiones de tuning	Características comunes
Definición de cuestiones	PB-0-004	PB-0-008	Estos requisitos están referidos al detalle que deben contemplar tanto las cuestiones de auditoría como las cuestiones de tuning. Su aplicación práctica implicará el desarrollo de estructuras que representen sus componentes.
Gestión de cuestiones	PB-0-005	PB-0-009	Tanto sobre las cuestiones de auditoría como sobre las cuestiones de tuning se requiere la implementación de funcionalidades de gestión que incluyen el alta, la modificación y la baja de ambos tipos de cuestiones.

Tabla 51.Relación entre requisitos identificados de cuestiones de auditoría y cuestiones de tuning.

La finalidad de llevar a cabo la agrupación de requisitos ha sido la de simplificar el proceso de desarrollo, intentando paralelizar la implementación de funcionalidades con características comunes. La siguiente tabla describe las tareas que se han llevado a cabo para cada agrupación de requisitos.

ID	Nombre	Tareas asociadas
PB-0-004 y PB-0-008	Definición de cuestiones	1. Inclusión en el modelo de datos de las cuestiones de auditoría y tuning 2. Elaboración de clases TO y FORM utilizadas para manipular las cuestiones de auditoría y tuning.
PB-0-005 y PB-0-009	Gestión de cuestiones	1. Planteamiento del requisito utilizando un diagrama de casos de uso. 2. Carga de datos en base de datos para permitir la realización de pruebas sobre cada una de las opciones. 3. Construcción de las opciones de "Alta de cuestiones de auditoría" y "Alta de cuestiones de tuning". 4. Prueba de la opciones de "Alta de cuestiones de auditoría" y "Alta de cuestiones de tuning". 5. Construcción de las opciones de "Modificación de cuestiones de auditoría" y "Modificación de cuestiones de tuning" en la aplicación. 6. Prueba de la opciones de "Modificación de cuestiones de auditoría" y "Modificación de cuestiones de tuning" en la aplicación. 7. Construcción de opciones de "Baja de cuestiones de auditoría" y "Baja de cuestiones de tuning". 8. Prueba de opciones de "Baja de cuestiones de auditoría" y "Baja de cuestiones de tuning".

Tabla 52.Relación entre requisitos y tareas identificadas en Sprint 2.

6.5.2 Elementos técnicos a destacar

En este punto aparecen aquellos elementos que se han utilizado para resolver determinados problemas y que se consideran particularmente relevantes en la realización del Sprint 2.

6.5.2.1 Inyección de dependencias

La inyección de dependencias [Wal11] es una técnica que permite facilitar la programación haciendo el código más fácil de comprender, reutilizar y probar. Cualquier aplicación con cierto grado de complejidad está formada por dos o más clases que colaboran entre sí para llevar a cabo algún tipo de lógica. Tradicionalmente, cada objeto es responsable de obtener sus propias referencias a los objetos con los que colabora (sus dependencias). Esto puede generar código muy acoplado y difícil de probar.

En la siguiente figura puede observarse un fragmento de código procedente de la clase SeccionController perteneciente a la aplicación AAS11:

```
@Controller
public class SeccionController {

        private SeccionService seccionService;

        public SeccionService getSeccionService() {
            return seccionService;
        }

        @Inject
        public void setSeccionService(SeccionService seccionService) {
            this.seccionService = seccionService;
        }
...
}
```

Figura 76.Ejemplo de Inyección de dependencias en clase SeccionController.

Centrándonos en el método setSeccionService perteneciente a la clase SeccionController lo primero que podemos observar es que no se realiza una operación new sobre un objeto del tipo SeccionService por lo que se supone la preexistencia de un objeto de este tipo (pero su creación no ha sido delegada en este método). SeccionService, adicionalmente, está implementado como un interfaz por lo que un objeto del tipo SeccionController no está acoplado con ninguna implementación específica de SeccionService. La principal ventaja que ofrece la inyección de dependencias es permitir un acoplamiento débil.

6.5.2.2 Eliminación de código reutilizable con plantillas

Uno de los elementos que ha facilitado en gran medida el desarrollo de la aplicación AAS11 y que a su vez constituye una de las ventajas que suministra el framework de Spring es la utilización de plantillas. La utilización de plantillas permite eliminar la necesidad de reescribir continuamente código reutilizable[98]. Un ejemplo claro de código reutilizable aparece cuando se trabaja con JDBC para realizar acciones sobre la base de datos. A continuación podemos ver un fragmento de código en el que se realiza la consulta de los datos asociados a un apartado relativo a una cuestión de tuning en el que no se utilizan plantillas:

[98] El código reutilizable [Wal11] es aquel que tenemos que escribir una y otra vez para llevar a cabo tareas habituales y sencillas.

```
public ApartadoTO consultarPorClave(int clave) {
                Connection con= null;
                PreparedStatement stmt= null;
                ResulSet resulSet=null;
                try {
                  conn = dataSource.getConnection();
                  stmt =  conn.preparedStatement(
                                "select * from APARTADOS where IDAPARTADO = ?");
                  stmt.setInt(1,clave);
                  resulSet =stmt.executeQuery();
                   ApartadoTO apartadoTO = null;
                  If (resulSet.next()){
                      apartadoTO=new ApartadoTO();
                      apartadoTO.setIdApartado(resultSet.getInt("IDAPARTADO"));
                      apartadoTO.setTituloApartado(resultSet.getString("TITULOAPARTADO"));
                      apartadoTO.setDesApartado(resultSet.getString("DESAPARTADO"));
                  }
                  return apartadoTO;
                } catch (SQLException e) {

                } finally {
                        if (resulSet!= null) {
                          try {
                            resulSet.close();
                          } catch (SQLException e) {}
                        }

                        if (stmt!= null) {
                          try {
                            stmt.close();
                          } catch (SQLException e) {}
                        }

                        if (conn!= null) {
                          try {
                            conn.close();
                          } catch (SQLException e) {}


                }
                 return null;
        }
```

Figura 77.Ejemplo de consulta habitual utilizando JDBC.

El ejemplo de la figura 77 muestra una consulta sobre la tabla APARTADOS para obtener todos los datos asociados a un determinado apartado de una cuestión de tuning. Como podemos observar, el código que realiza toda la consulta se encuentra rodeado de código JDBC que implica la realización de las acciones de creación de conexión, creación de sentencia y la acción de ejecución de la sentencia.

Inherente a la utilización de JDBC, se necesita capturar SQLException para, al menos, tener constancia de los posibles fallos producidos durante el proceso. Una vez realizadas todas las acciones habría que cerrar el objeto de conexión, el objeto que representa la sentencia y objeto asociado al conjunto de resultados. Cualquier operación utilizando JDBC requeriría hacer uso de la misma estructura por lo que esta parte de código común se considera código reutilizable. Spring permite eliminar este código reutilizable a encapsulándolo en plantillas. *JdbcTemplate* [Wal11] hace posible la llevar a cabo operaciones de base de datos sin todos los requisitos asociados con el JDBC tradicional. En la siguiente figura aparece el método consultarPorClave utilizado en la aplicación AAS11 en el que se utiliza la plantilla *JdbcTemplate*:

```
public ApartadoTO consultarPorClave(int clave) {
    JdbcTemplate jdbcTemplate = this.getJdbcTemplate();
    String consulta = "select * from APARTADOS where IDAPARTADO = ?";
    String [] parametros = {clave};
    ApartadoTO apartadoTO = jdbcTemplate.query(consulta, parametros, new
        ResultSetExtractor<ApartadoTO>() {
            public ApartadoTO extractData(ResultSet resultSet) throws
             SQLException, DataAccessException {
                if (resultSet.next()) {
                    apartadoTO=new ApartadoTO();
                    apartadoTO.setIdApartado(resultSet.getInt("IDAPARTADO"));
                     apartadoTO.setTituloApartado(
                                resultSet.getString("TITULOAPARTADO"));
                     apartadoTO.setDesApartado(
                                resultSet.getString("DESAPARTADO"));
                        return apartadoTO;
                 }
                else {
                      return null;
                }
            }
        });
        return apartadoTO;
}
```

Figura 78.Ejemplo de utilización de plantillas en consulta de base de datos.

Cómo se puede observar, el método de la figura 78 implementa una solución más sencilla centrada en la selección de los datos de un usuario particular de la aplicación AAS11. Todo el código reutilizable JDBC está oculto en la plantilla reduciendo la complejidad de la implementación.

6.5.2.3 Implementación de la opción de paginación

La paginación constituye una funcionalidad utilizada en múltiples puntos de la aplicación AAS11. Siempre que aparece un listado, habrá una referencia explícita a esta funcionalidad. En la figura que aparece a continuación podemos observar la clase que permite implementar la opción de paginación:

```
package org.bastanchu.pfc.aas11.util;

import java.util.ArrayList;
import java.util.Collection;

public class Paginador<T> {

	public static final int ITEMS_POR_PAGINA_POR_DEFECTO = 9;

	private Collection<T> datos;
	private int itemsPorPagina;

	public Paginador(Collection<T> datos, int itemsPorPagina) {
		this.datos = datos;
		this.itemsPorPagina = itemsPorPagina;
	}

	public Paginador(Collection<T> datos) {
		this(datos, ITEMS_POR_PAGINA_POR_DEFECTO);
	}

	public int getNumPaginas() {
		return ((this.datos.size() - 1)/ this.itemsPorPagina) + 1;
	}

	public Collection<T> getItemsEnPagina(int numPag) {
		Collection<T> items = new ArrayList<T>();
		Object []itemsArray = datos.toArray();
		int limiteInferior = numPag * itemsPorPagina;
		int limiteSuperior = limiteInferior + itemsPorPagina;
		for(int i = limiteInferior ; (i < datos.size()) && (i < limiteSuperior) ; i++) {
			items.add((T) itemsArray[i]);
		}
		return items;
	}
}
```

Figura 79.Clase "Paginador" utilizada para implementar la opción de paginación.

Lo primero que se puede observar en la clase Paginador es que se ha intentado construir una implementación lo más general posible de esta clase, para lo que se han utilizado los genéricos que habilita el lenguaje de programación JAVA con la finalidad de no asociar un tipo de datos explicito a la implementación de esta opción. Lo siguiente que podemos observar es la declaración de atributos. La clase Paginador contiene un atributo que identifica la colección de páginas correspondiente (*datos*) y un atributo que permite determinar el número de elementos por página (*itemsPorPagina*). Adicionalmente, se define una constante que establece el número de elementos por página por defecto (*ITEMS_POR_PAGINA_POR_DEFECTO*). A continuación, aparecen los constructores asociados a la clase. El primero de ellos permite especificar un número determinado de elementos por página, mientras que el segundo utiliza el valor de la constante para establecer dicho número.

Finalmente aparecen los métodos (*getNumPaginas*) y (*getItemsEnPagina*). El primer método permite obtener el número de páginas asociadas al elemento sometido a la acción de paginación mientras que el segundo método permite acotar aquellos elementos de la colección que pertenecen a una determinada página.

Un ejemplo de utilización de esta clase puede verse en la siguiente figura, en la que se establece la paginación para las secciones que permiten la clasificación de las cuestiones de auditoría:

```
// Listar secciones
@PreAuthorize("hasRole('6')")
@RequestMapping("/listaSecciones")
public ModelAndView obtenerListaSecciones(SeccionForm formEntrada) {
        Map<String, Object> model = new HashMap<String, Object>();
        List<SeccionTO> listaSecciones =
            this.getSeccionService().obtenerTodasLasSecciones();
        Paginador<SeccionTO> seccionesPaginadas = new
             Paginador<SeccionTO>(listaSecciones,6);
        model.put("lista", seccionesPaginadas);
        model.put("command", formEntrada);
        return new ModelAndView("listaSecciones", model);
}
```

Figura 80.Utilización de la clase "Paginador" en la lista de secciones.

En esta figura podemos observar como se define el objeto *seccionesPaginadas* utilizando la clase Paginador particularizada para objetos del tipo *SeccionTO*. En este caso se utiliza el constructor que permite definir el número de elementos por página. Finalmente, el contenido de esta lista será el que se visualice en la pantalla que presenta el listado de secciones de auditoría.

Si pensamos en la implementación de la funcionalidad de paginación podríamos considerar que teóricamente cumple todos los requisitos para ser considerada como un aspecto. La programación orientada a aspectos [Wal11] suele definirse como una técnica que promueve la separación de problemas dentro de un sistema software. Los sistemas están formados por varios componentes, cada uno de los cuales es responsable de una funcionalidad específica. A menudo estos componentes cuentan con responsabilidades adicionales más allá de sus funciones básicas. Los servicios del sistema, como inicio de sesión, administración de transacciones y seguridad, suelen encontrarse en componentes cuya responsabilidad principal corresponde a otros elementos. A estos elementos se les suele denominar preocupaciones transversales, ya que tienden a incluirse en diferentes componentes del sistema.

Al repartir estas preocupaciones transversales entre diferentes componentes, se introducen dos niveles de complejidad en el código:

- El código que implementa la denominada "preocupación transversal" a nivel de sistema se duplica entre los diferentes componentes. Esto quiere decir que, si quiere modificar la forma en que funcionan esas preocupaciones, tendrá que acceder a varios componentes. Incluso, aunque se haya abstraído la preocupación en un módulo aparte, para que el impacto de sus componentes sea una única ejecución, la ejecución del método se duplica en varias ubicaciones.

- Sus componentes van a incluir código innecesario que no va a estar alineado con su funcionalidad básica. Un método para añadir una entrada en una libreta de direcciones sólo debería encargarse de cómo añadir la dirección, y no de si es seguro o transaccional.

La programación orientada a aspectos (AOP) habilita mecanismos para incluir en módulos estos servicios y, a continuación, aplicarlos de forma declarativa a los componentes a los que deben afectar. Esto permite que los componentes tengan una mayor cohesión entre sí y se centren en sus objetivos concretos, ignorando el resto de servicios de sistema que pueden verse implicados.

En el anterior párrafo es en el que se menciona una de las características de la programación orientada a aspectos que no cumple la implementación de la funcionalidad de paginación. En concreto, la utilización de esta implementación no se está realizando de forma declarativa sino que se está haciendo de forma explícita, con lo que se aumenta el acoplamiento[99].

6.5.3 Reunión de revisión y retrospectiva del Sprint 2

En esta reunión se realiza un análisis del resultado del Sprint llevado a cabo. Como resultado se concluye que no se han encontrado particulares dificultades en la implementación de estas opciones de gestión, debido a que ya existe el precedente de la implementación de la gestión de usuarios.

A continuación se muestra el diagrama Burn-Down asociado al Sprint 2:

[99] El framework de Spring dota de mecanismos que permiten declarar determinadas clases cómo aspectos y asociar dichos aspectos con las clases sobre las que se aplican a través de ficheros de configuración XML.

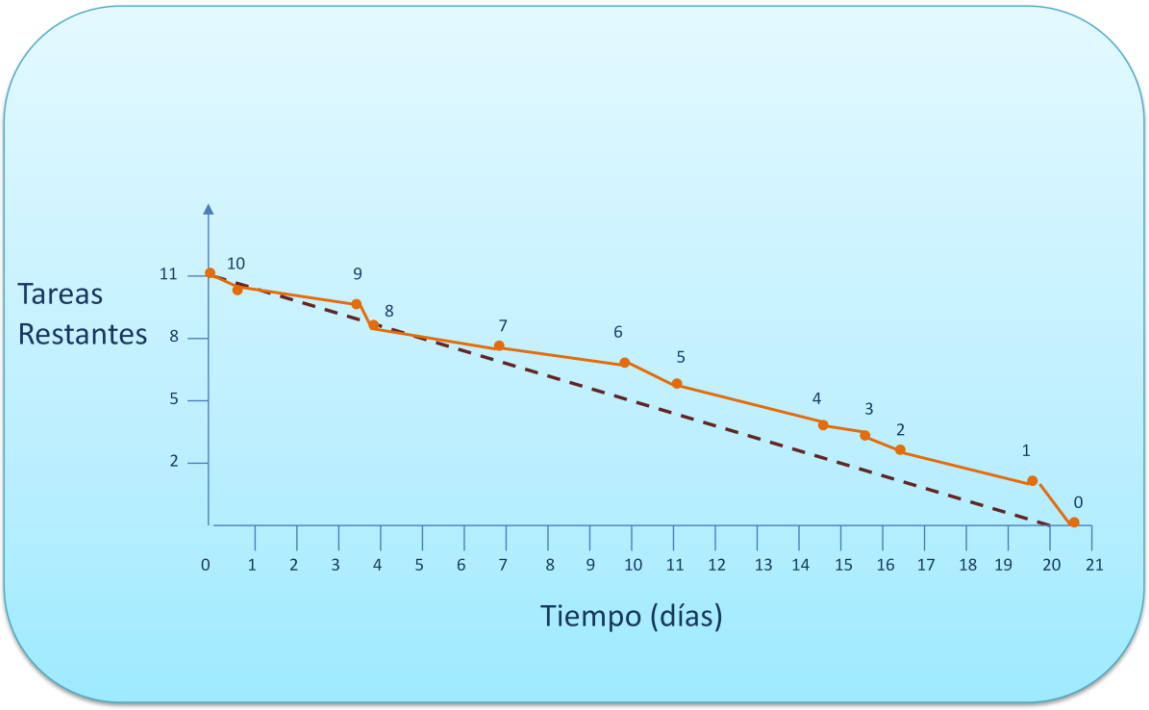


Figura 81.Diagrama Burn-Down asociado al Sprint 2.

En este diagrama se puede observar que se ha producido una desviación de 0,75 días. Aunque la planificación de la implementación de las opciones de gestión asociadas a las cuestiones de tuning y auditoría estaba basada en la implementación de las opciones de gestión de usuarios, el retraso se ha producido en esta ocasión al introducir un volumen de datos significativo para hacer una prueba exhaustiva del sistema.

6.6 Síntesis del Sprint 3

6.6.1 Reunión de planificación del Sprint 3

En la reunión de planificación asociada al Sprint 3 se seleccionan los siguientes requisitos para su implementación:

- PB-0-006 Clasificación de cuestiones de auditoría en secciones: este requisito establece la necesidad de disponer de una clasificación que permita agrupar las cuestiones de auditoría en clases para facilitar su gestión. Adicionalmente, se implementarán opciones que permitan gestionar dichas clases (altas, bajas y modificaciones de secciones).

- PB-0-007 Configuraciones particularizadas de cuestiones de auditoría: una vez que se han implementado las opciones que hacen referencia a la gestión de cuestiones de auditoría y a su administración a través de secciones, será necesario disponer de algún mecanismo que permita llevar a cabo selecciones personalizadas de estas cuestiones de auditoría con la finalidad de dotar a la aplicación de mecanismos de personalización utilizables por cada uno de sus usuarios.

- PB-0-010 Clasificación de cuestiones de tuning en apartados: este requisito hace referencia a la necesidad de disponer de un elemento que permita establecer una clasificación sobre las diferentes cuestiones de tuning almacenadas en la aplicación AAS11. Asociado a la implementación de este requisito se dispondrá de mecanismos que permitan llevar a cabo la gestión de estos apartados.

- PB-0-011 Configuraciones particularizadas de cuestiones de tuning: la traducción de este requisito será disponer de una opción que permita a los diferentes usuarios de la aplicación llevar a cabo una selección de aquellas cuestiones de auditoría que necesiten en sus informes.

En la siguiente tabla se muestra el Sprint Backlog asociado al Sprint 3:

ID	Nombre	Prioridad	Coste	Completado	Sprint 1	Sprint 2	Sprint 3
PB-0-006	Clasificación de cuestiones de auditoría en secciones	0.6	10	0	0	0	0
PB-0-007	Configuraciones particularizadas de cuestiones de auditoría	0.5	30	0	0	0	0
PB-0-010	Clasificación de cuestiones de tuning en apartados	0.6	10	0	0	0	0
PB-0-011	Configuraciones particularizadas de cuestiones de tuning	0.5	30	0	0	0	0

Tabla 53.Sprint Backlog 3.

Al repasar la definición de estos requisitos podemos identificar elementos comunes en su definición. Estos elementos comunes se describen en la siguiente tabla:

Contenido del requisito	Cuestiones de auditoría	Cuestiones de tuning	Características comunes
Clasificación de cuestiones	PB-0-006	PB-0-010	Estos requisitos hacen referencia a las clasificaciones asociadas a las cuestiones de auditoría y a las cuestiones de tuning. Estas clasificaciones implicarán la implementación de opciones que permitan su gestión (altas bajas y modificaciones).
Configuraciones particularizadas de cuestiones para cada usuario	PB-0-007	PB-0-011	Se debe dotar a la aplicación de mecanismos que permitan la selección particularizada de las cuestiones de auditoría y tuning que compondrán un test. En ambos casos se mostrará una lista con cuestiones seleccionables.

Tabla 54.Relación entre requisitos identificados de cuestiones de auditoría y cuestiones de tuning.

Al igual que en el caso del Sprint 2, está agrupación de funcionalidades ha permitido identificar características comunes que permitirán simplificar el proceso de implementación. En la siguiente tabla se muestra la relación establecida entre cada agrupación de requisitos y sus tareas asociadas:

ID	Nombre	Tareas asociadas
PB-0-006 y PB-0-010	Clasificación de cuestiones	1. Planteamiento del requisito utilizando un diagrama de casos de uso. 2. Introducción en el modelo de taos de la clasificación de cuestiones de auditoría y la clasificación de cuestiones de tuning. 3. Carga de datos en la base de datos para la realización de pruebas posteriores. 4. Construcción de las opciones de "Alta de secciones" y "Alta de apartados" 5. Prueba de las opciones de "Alta de secciones" y "Alta de apartados". 6. Construcción de las opciones "Modificación de secciones" y "Modificación de apartados". 7. Prueba de las opciones "Modificación de secciones" y "Modificación de apartados". 8. Construcción de las opciones de "Baja de secciones" y "Baja de apartados". 9. Prueba de las opciones de "Baja de secciones" y "Baja de apartados". 10. Prueba de opciones de "Baja de secciones" y "Baja de apartados". 11. Utilización de Hibernate Validator en los formularios ya implementados.
PB-0-007 y PB-0-011	Configuraciones particularizadas de cuestiones para cada usuario	1. Planteamiento del requisito utilizando un diagrama de casos de uso. 2. Introducción en el modelo de datos de la selección personalizada de cuestiones de auditoría y tuning. 3. Construcción de las opciones de "Selección de cuestiones de auditoría" y "Selección de cuestiones de tuning". 4. Prueba de opciones de "Selección de cuestiones de auditoría" y "Selección de cuestiones de tuning".

Tabla 55.Relación entre requisitos y tareas identificadas en Sprint 3.

6.6.2 Elementos técnicos a destacar

El elemento técnico que se destaca durante la realización del Sprint 3 es la incorporación de Hibernate Validator. En este punto se expondrá el proceso de incorporación de esta tecnología en la aplicación AAS11 y ejemplos de su utilización.

6.6.2.1 Utilización de Hibernate Validator

La validación de datos es una tarea común que se lleva a cabo dentro de cualquier aplicación, desde la capa de presentación a la capa de persistencia. A menudo, la misma lógica de validación se lleva a cabo en cada capa, lo que consume mucho tiempo y constituye un proceso propenso a errores.

Para evitar la duplicación de estas validaciones en cada capa, los desarrolladores suelen agrupar lógica de validación directamente en el modelo de dominio[100], lo que sobrecarga las clases asociadas al modelo de dominio con código de validación.

JSR 303 - Bean Validation - define un modelo de metadatos y un API para la validación de entidades. El origen de metadatos predeterminado está constituido por anotaciones, con la posibilidad de anular y extender estos metadatos mediante el uso de XML. Este API no está ligado a ninguna capa de aplicación específica ni a ningún modelo de programación predefinido. Este API no está ligado explícitamente ni a la capa web ni a la capa de persistencia, y está disponible tanto para la programación de aplicaciones del lado del servidor como para aplicaciones ricas desarrolladas en el lado del cliente.

El primer paso para incorporar Hibernate Validator al proyecto AAS11 ha sido la modificación del fichero pom.xml incluyendo los siguientes elementos:

```
<dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-validator</artifactId>
        <version>4.3.0.Final</version>
</dependency>
```

Figura 82.Inclusión de Hibernate Validator en fichero Pom.xml.

El siguiente paso ha sido la modificación del fichero de configuración aas11-servlet.xml con la finalidad de incluir una referencia al fichero de claves utilizado para realizar las validaciones sobre los campos de los formularios:

```
<!-- Configuracion de archivo de claves -->
<bean id="messageSource"
  class="org.springframework.context.support.
  ReloadableResourceBundleMessageSource">
  <property name="basenames">
        <list>
                <value>classpath:messages</value>
                <value>classpath:ValidationMessages</value>
        </list>
  </property>
  <property name="defaultEncoding" value="ISO-8859-1"/>
</bean>
```

Figura 83.Referencia a fichero de claves utilizado por Hibernate Validator en aas11-servlet.xml.

[100] Un modelo del dominio [Lar02] es una representación de las clases conceptuales del mundo real, no de componentes software. No se trata de un conjunto de diagramas que describen clases software, u objetos software con responsabilidades. Un modelo del dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. También se les denomina modelos conceptuales, modelo de objetos del dominio y modelos de objetos de análisis.

Una vez establecida la configuración para utilizar Hibernate-Validator procedemos a incorporar las anotaciones en las clases form utilizadas para intercambiar información con la capa de presentación:

```
package org.bastanchu.pfc.aas11.form;

import javax.validation.constraints.DecimalMin;
import javax.validation.constraints.Digits;
import javax.validation.constraints.Min;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;

import org.hibernate.validator.constraints.NotEmpty;

public class SeccionForm {
	private int numPaginaListado;
	@Digits(integer=6, fraction=0, message= "{DecimalMin.seccionForm.idSeccion}")
	private int idSeccion;
	@NotEmpty(message = "{NotEmpty.seccionForm.tituloSeccion}")
	private String tituloSeccion;
	@NotEmpty(message = "{NotEmpty.seccionForm.desSeccion}")
	private String desSeccion;

	public int getNumPaginaListado() {
		return numPaginaListado;
	}
	public void setNumPaginaListado(int numPaginaListado) {
		this.numPaginaListado = numPaginaListado;
	}
	public int getIdSeccion() {
		return idSeccion;
	}
	public void setIdSeccion(int idSeccion) {
		this.idSeccion = idSeccion;
	}
	public String getTituloSeccion() {
		return tituloSeccion;
	}
	public void setTituloSeccion(String tituloSeccion) {
		this.tituloSeccion = tituloSeccion;
	}
	public String getDesSeccion() {
		return desSeccion;
	}
	public void setDesSeccion(String desSeccion) {
		this.desSeccion = desSeccion;
	}
}
```

Figura 84.Clase SeccionForm en la que se utilizan las anotaciones de Hibernate Validator.

En la implementación de la clase SeccionForm podemos observar la utilización de las anotaciones en la definición de los atributos. De esta forma, podemos observar como de manera declarativa, se hace referencia a las restricciones que se imponen sobre cada uno de los atributos. Así, el atributo *idSeccion* aparece limitado, utilizando la anotación *@Digits*, a un entero de seis dígitos sin decimales. En el caso de que se incumplan estas restricciones se avisará a través del mensaje identificado por la clave *DecimalMin.seccionForm.idSeccion*. En los atributos *tituloSeccion* y *desSeccion* se establece la restricción de "no vacío" a través del uso de la anotación *@NotEmpty*.

Finalmente, podemos observar como se utilizan las anotaciones de Hibernate Validator en la clase Controller que lleva a cabo la acción de alta de una sección:

```
@PreAuthorize("hasRole('5')")
@RequestMapping("/altaSeccion")
public ModelAndView altaSeccion(@ModelAttribute("command") @Valid
        SeccionForm form, BindingResult result) {
        if (result.hasErrors()){
                ModelAndView modelAndView = new
                ModelAndView("modificacionSeccion");
                modelAndView.addObject("command", form);
                return new ModelAndView("altaSeccion");
        }else{
                SeccionTO seccionTO =
                this.getSeccionService().consultarSeccion(form.getIdSeccion());
                if (seccionTO != null) {
                        // La sección ya existe, se notifica un error
                        result.addError(new FieldError("seccion", "idSeccion",
                        "La sección ya existe"));
                        ModelAndView modelAndView = new
                        ModelAndView("altaSeccion");
                        modelAndView.addObject("command", form);
                        return modelAndView;
                }
                else {
                        SeccionTO seccion = this.deFormSeccionTO(form);
                        this.getSeccionService().insertarSeccion(seccion);
                        return new
                        ModelAndView("seccionInsertadaCorrectamente");
                }
        }
}
```

Figura 85.Método altaSeccion perteneciente a la clase SeccionController.

En la implementación del método *altaSeccion* podemos ver la utilización de la anotación @Valid asociada al atributo form. Esta anotación permite utilizar las validaciones definidas en la clase SeccionForm. El objeto result se utiliza para determinar si se han producido errores en el contenido de los campos del formulario. Finalmente, podemos observar que el caso de clave duplicada en la inserción de una sección ha sido tratado de forma explícita en este método.

6.6.3 Reunión de revisión y retrospectiva del Sprint 3

En esta reunión se realiza un análisis del Sprint llevado a cabo. En el Sprint 3 no se han producido desviaciones con respecto a la estimación inicialmente realizada. El punto más destacado de este Sprint ha sido la incorporación de Hibernate Validator y su extensión a los formularios implementados hasta este momento.

En la siguiente figura se puede observar el diagrama Burn Down asociado al Sprint 3:

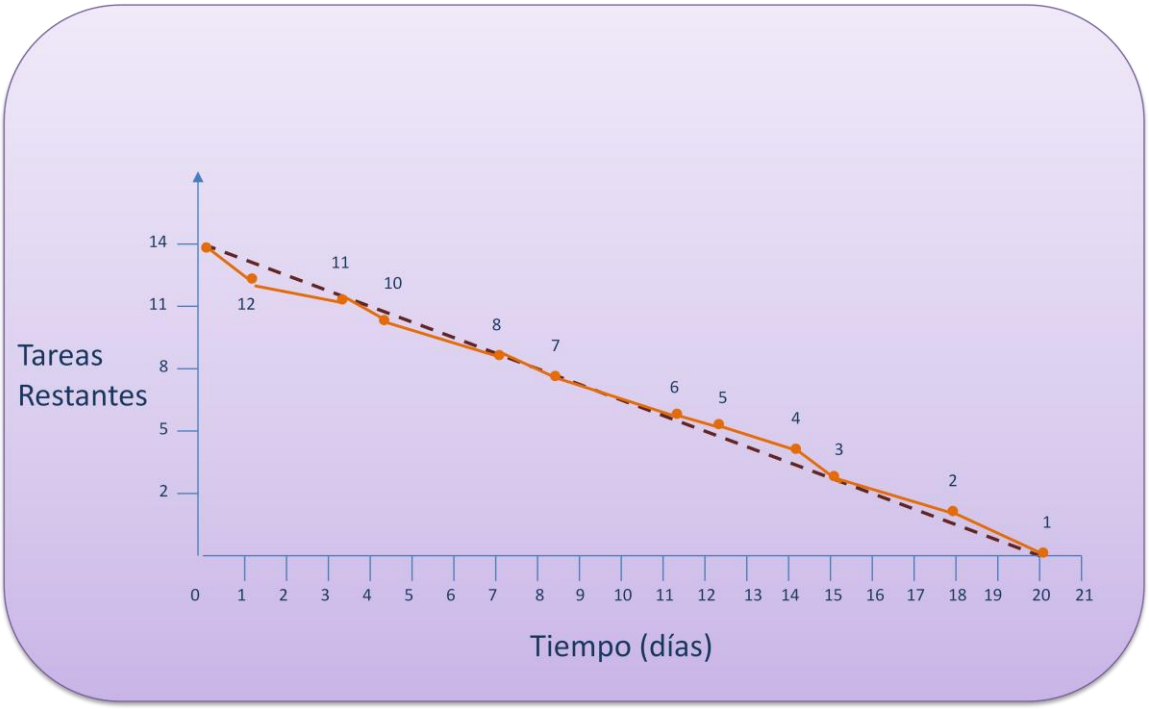


Figura 86.Diagrama Burn-Down asociado al Sprint 3.

En este diagrama se puede observar que las tareas se han distribuido de forma más homogénea y que en ningún instante se han producido grandes desviaciones con respecto a la estimación inicialmente realizada.

6.7 Síntesis del Sprint 4

6.7.1 Reunión de planificación del Sprint 4

En la reunión de planificación asociada al Sprint 4 se seleccionan los siguientes requisitos para su implementación:

- PB-0-012 Elaboración de test: uno de los objetivos principales de la aplicación es permitir la realización de distintos test a los usuarios para que puedan evaluar un determinado sistema. La implementación de este requisito exige consultar la información que gestiona la aplicación AAS11 para visualizar cada una de las preguntas de test realizadas a los usuarios.

- PB-0-013 Gestión de test: una vez que los usuarios han realizado los test en la aplicación, se permitirá su almacenamiento y consulta para que el propio usuario pueda llevar a cabo la gestión de sus test.

- PB-0-014 Informes en formato Word: este requisito establece la necesidad de que los informes generados por la aplicación se encuentren codificados en un formato que sea manipulable por parte de los usuarios.

En la siguiente tabla se muestra el Sprint Backlog asociado al Sprint 4:

ID	Nombre	Prioridad	Coste	Completado	Sprint 1	Sprint 2	Sprint 3	Sprint 4
PB-0-012	Elaboración de Test	0.6	10	0	0	0	0	0
PB-0-013	Gestión de Test	0.5	30	0	0	0	0	0
PB-0-014	Informes en formato Word	0.6	10	0	0	0	0	0

Tabla 56.Sprint Backlog 4.

6.7.2 Elementos técnicos a destacar

Los elementos técnicos a destacar utilizados durante la implementación del Sprint 4 se enuncian a continuación:

- Librería POI (generación de documentos en formato Word): la librería Apache POI [POI13] se utiliza para manipular y dar formato a documentos asociados al paquete ofimático Microsoft Office.

- Utilización de la librería JFreechart en la generación de gráficos: JFreechart [Dan13] es una librería de código abierto disponible para Java que permite a los usuarios generar gráficos y diagramas fácilmente.

- Almacenamiento de de fichero en una columna de tipo BLOB en bbdd: el documento obtenido como resultado de la realización de cada test de auditoría y tuning, será almacenado en la base de datos en una columna de tipo BLOB para su posterior recuperación.

6.7.2.1 Utilización de librería para Word POI

Con la finalidad de utilizar un formato de salida de documento de Word, ha sido necesaria la incorporación de la librería Apache POI. Esta librería [POI13] contiene un conjunto de API's que permiten manipular varios formatos de ficheros basados en OOXML y en el formato de composición de documentos Microsoft's OLE2. Los pasos que se han llevado a cabo para manipular este tipo de documentos se describen a continuación:

- Elaboración de una plantilla de documento: el primer paso ha sido diseñar una plantilla de documento Word inicial sobre la que se irá incluyendo el contenido dinámico de cada test realizado. Las siguientes figuras muestran el detalle de la plantilla utilizada como base. En esta plantilla se puede observar como se han definido una serie de etiquetas especiales que permitirán identificar tanto el título del informe (${tituloDelInforme}) como la fecha (${fecha}).

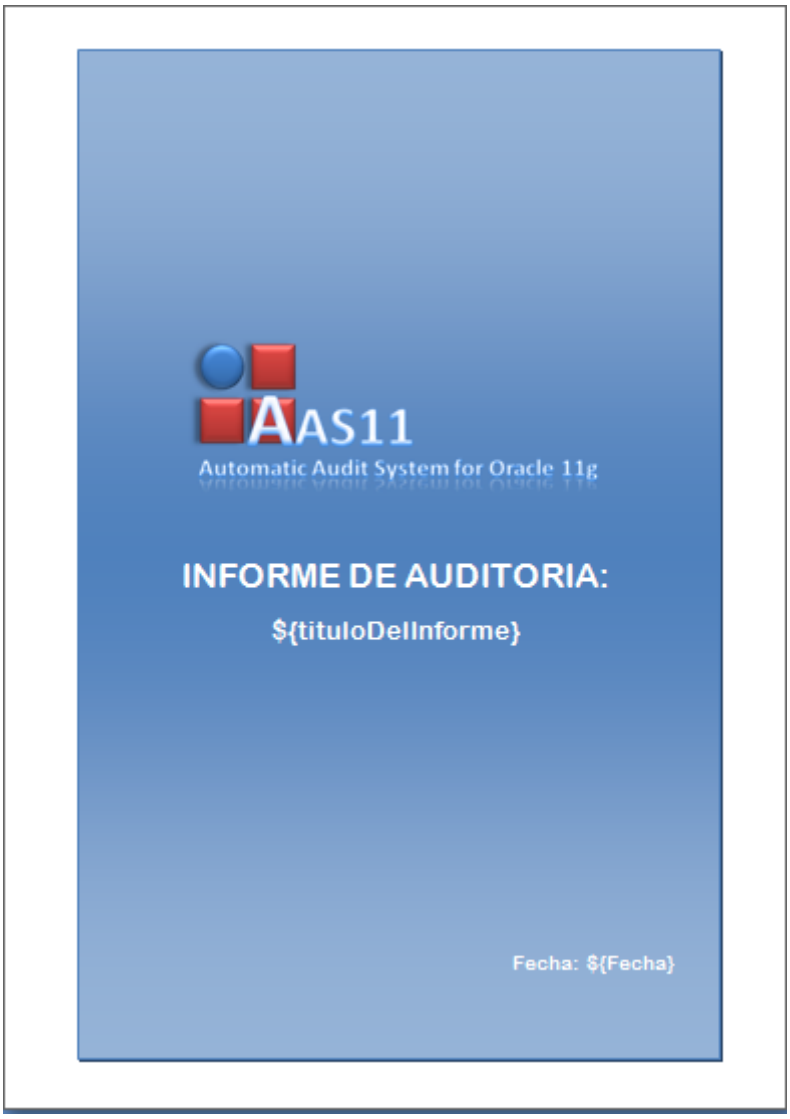


Figura 87. Inicio del documento generado automáticamente por AAS11.

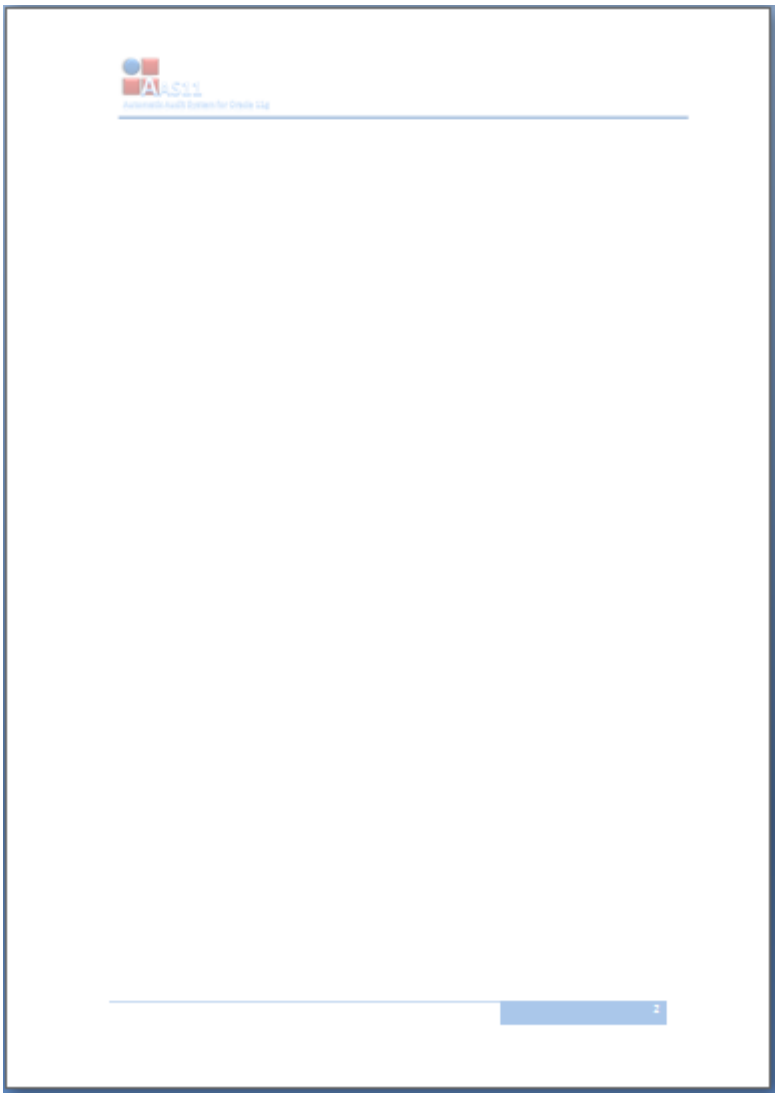


Figura 88. Contenido del documento en el que inicialmente se ha definido una cabecera y un pié.

- Implementación de la clase InformeWordServiceImpl: esta clase implementará el conjunto de funcionalidades relacionadas con la manipulación de este tipo de documentos. Para construir un documento Word se utilizan las siguientes clases pertenecientes a la librería POI:

    - XWPFDocument: esta clase se utiliza para representar el documento que va a ser tratado.
    - XWPFParagraph: esta clase se utilizará para hacer referencia a cada uno de los párrafos en los que se descompone el documento.
    - XWPFRun: clase utilizada para representar una región de texto con un conjunto de características comunes.

Una vez expuestas las clases de base que se utilizan para el tratamiento de un documento de este tipo, se muestra en las siguientes figuras el detalle de algunos de los métodos pertenecientes a dicha clase:

```
public XWPFDocument generarInformeWord(InformeForm informeForm)
throws IOException {
        InputStream stream = Thread.currentThread().getContextClassLoader().
         getResourceAsStream(PLANTILLA_INFORME);
        XWPFDocument document = new XWPFDocument(stream);
        establecerTituloDeInforme(document, informeForm.getIdInforme());
        establecerFecha(document, informeForm.getFechaInforme());
        anyadirCuestiones(document, informeForm);
        stream.close();
        return document;
}
```

Figura 89.Método generaInformeWord perteneciente a la clase InformeWordServiceImpl.

En el método generaInformeWord se puede observar que la composición de un informe consiste en: establecer el título del informe, establecer la fecha y añadir las cuestiones (auditoría y tuning) de las que se compone el informe.

```
private void establecerTituloDeInforme(XWPFDocument document, String titulo) {
        final String TITULO_DEL_INFORME = "${tituloDelInforme}";
        for (XWPFParagraph parrafo : document.getParagraphs()) {
                for (XWPFRun racha : parrafo.getRuns()) {
                        if ((racha.getText(0) != null) &&
                        racha.getText(0).contains(TITULO_DEL_INFORME)) {
                                racha.setText(racha.getText(0).
                                replace(TITULO_DEL_INFORME, titulo), 0);
                        }
                }
        }
}
```

Figura 90.Método establecerTituloDelInforme perteneciente a la clase InformeWordServiceImpl.

En el método establecerTituloDelInforme se aprecia en qué consiste esta acción. Lo que se realiza, es una búsqueda párrafo a párrafo y por cada párrafo, racha a racha. Por cada racha, se comprueba la existencia de la cadena que representa el título del informe. En el caso de que se encuentre, se substituye por su valor correspondiente.

```
private void establecerFecha(XWPFDocument document, Date fecha) {
        final String FECHA = "${Fecha}";
        DateFormat formatoFecha = new SimpleDateFormat("dd/MM/yyyy");
        String fechaFormateada = formatoFecha.format(fecha);
        for (XWPFParagraph parrafo : document.getParagraphs()) {
                for (XWPFRun racha : parrafo.getRuns()) {
                        if ((racha.getText(0) != null) &&
                        racha.getText(0).contains(FECHA)) {
                                racha.setText(racha.getText(0).replace(FECHA,
                                fechaFormateada), 0);
                        }
                }
        }
}
```

Figura 91.Método establecerFecha perteneciente a la clase InformeWordServiceImpl.

Al igual que en el método establecerTituloDelInforme en el método establecerFecha se realiza una búsqueda párrafo a párrafo y por cada párrafo, racha a racha. Por cada racha, se comprueba la existencia de la cadena que representa la fecha del informe. En el caso de que se encuentre, se substituye por el valor previamente establecido.

```
private void anyadirUnaCuestion(XWPFDocument document, CuestionTO unaCuestion,
boolean respuestaCuestion) {
        // Nombre de la cuestion
        XWPFParagraph parrafoNombre =  document.createParagraph();
        XWPFRun rachaNombre =  parrafoNombre.createRun();
        rachaNombre.setFontFamily("Times New Roman");
        rachaNombre.setFontSize(14);
        rachaNombre.setBold(true);
        rachaNombre.setColor("000000");
        rachaNombre.setText(unaCuestion.getDesCuestion() + '\n', 0);
        //rachaNombre.addBreak();
        // Comentario de cuestion
        XWPFParagraph parrafoComentario =  document.createParagraph();
        XWPFRun rachaComentario =  parrafoComentario.createRun();
        rachaComentario.setFontFamily("Times New Roman");
        rachaComentario.setFontSize(12);
        rachaComentario.setItalic(true);
        rachaComentario.setColor("808080");
        rachaComentario.setText("Comentario: " +unaCuestion.getComenCuestion(), 0);
        //rachaComentario.addBreak();
        if ((unaCuestion.getSelCuestion() != null) &&
        !unaCuestion.getSelCuestion().trim().equals("")) {
                anyadirTablaConsulta(document, unaCuestion);
        }
        // Respuesta cuestion
        XWPFParagraph parrafoRespuesta =  document.createParagraph();
        XWPFRun rachaRespuesta=  parrafoRespuesta.createRun();
        rachaRespuesta.setFontFamily("Times New Roman");
        rachaRespuesta.setFontSize(14);
        rachaRespuesta.setColor("000000");
        XWPFParagraph parrafoBlanco =  document.createParagraph();
        XWPFRun rachaBlanco=  parrafoBlanco.createRun();
}
```

Figura 92.Método anyadirUnaCuestion perteneciente a la clase InformeWordServiceImpl.

En el método anyadirUnaCuestion se puede apreciar la división de párrafos y rachas establecida. Vemos que se define un párrafo y una racha para establecer el enunciado de la cuestión junto con el estilo a aplicar. A continuación, se utiliza un párrafo y una racha para establecer el comentario asociado a la cuestión. En este caso, si la cuestión dispone de una consulta asociada, se añade la tabla que resulta de realizar esta consulta. Finalmente, se establece un nuevo párrafo y una nueva racha para insertar el contenido de la respuesta asociado a la consulta. Al igual que en los casos anteriores, se define un estilo a aplicar sobre este contenido.

```
private void anyadirTablaConsulta(XWPFDocument document, CuestionTO
unaCuestion) {
        ConsultaGenericaService consultaGenericaService =
         getConsultaGenericaService();
        ResultadoConsultaGenericaTO resultadoConsulta = consultaGenericaService.
         realizarConsultaGenerica(unaCuestion.getSelCuestion());
        //Anyadimos la cabecera
        XWPFTable tabla = document.createTable();
        XWPFTableRow filaCabecera = tabla.getRow(0);
        for (int i = 0 ; i < resultadoConsulta.getColumnas().size() ; i++)
        {
                String tituloColumna = resultadoConsulta.getColumnas().get(i);
                XWPFTableCell celdaCabecera = null;
                if (i == 0) {
                        celdaCabecera = filaCabecera.getCell(0);
                }
                else {
                        celdaCabecera = filaCabecera.createCell();
                }
                XWPFParagraph parrafoCabecera =celdaCabecera.
                 getParagraphs().get(0);
                XWPFRun rachaCabecera = parrafoCabecera.createRun();
                rachaCabecera.setText(tituloColumna);
        }
        // Y despues el resto de filas
        for (Map<Integer,Object> mapFilaDatos : resultadoConsulta.getDatos()) {
                XWPFTableRow filaDatos = tabla.createRow();
                for (int i = 0 ; i < mapFilaDatos.size() ; i++) {
                        Object dato = mapFilaDatos.get(i);
                        XWPFTableCell celdaDatos = filaDatos.getCell(i);
                        if (dato != null) {
                                XWPFParagraph parrafoDatos =celdaDatos.
                                getParagraphs().get(0);
                                XWPFRun rachaDatos = parrafoDatos.createRun();
                                rachaDatos.setText(dato.toString());
                        }
                }
        }
}
```

Figura 93.Método anyadirTablaConsulta perteneciente a la clase InformeWordServiceImpl.

El método anyadirTablaConsulta se utiliza para añadir las tablas resultantes de la realización de consultas sobre la base de datos pertenecientes a aquellas cuestiones de auditoría y tuning que dispongan de esta opción. En este método se utilizan las clases XWPFTable, XWPFTableRow y XWPFTableCell que representan respectivamente la tabla a representar, las filas contenidas en la misma y cada una de las celdas que la componen. Como puede verse en el propio método, la composición de la tabla se realiza en dos fases. La primera fase lleva a

cabo la construcción de la cabecera y la segunda fase realiza la construcción de cada una de las filas que componen la tabla.

- Utilización de la opción de "generación de informes" en el Controller: en el Controller NuevoInformeController se define el método generarInforme que utiliza la clase previamente expuesta para materializar la generación del informe en formato Word:

```
@RequestMapping("/generarInforme.docx")
public ModelAndView generarInforme(InformeForm form, HttpServletResponse
response, HttpSession sesion) throws IOException {
        UsuarioAutenticadoSesionTO usuarioSessionTO =
        (UsuarioAutenticadoSesionTO)
        sesion.getAttribute("datosUsuarioSesion");
        String idUsuario = usuarioSessionTO.getUsuario();
        List<CuestionAuditoriaTO> listaCuestionesAuditoriaUsuario =
        this.getCuestionAuditoriaService().
        obtenerCuestionesDeAuditoriaPorUsuario(idUsuario);
        form.setCuestiones(listaCuestionesAuditoriaUsuario);
        response.setCharacterEncoding("application/vnd.openxmlformats-
        officedocument.wordprocessingml.document");
        XWPFDocument document =
        this.getInformeWordService().generarInformeWord(form);
        document.write(response.getOutputStream());
        return null;
}
```

Figura 94.Método generarInforme perteneciente a la clase NuevoInformeController.

En el método generarInforme se puede observar la recuperación de la las respuestas llevadas a cabo por el usuario asociadas al test que acaba de realizar. Tras esto, se establece el formato del documento que se va a generar. Finalmente, se procede a la generación del documento, lo que se materializará, en última instancia, en una ventana con el documento Word resultante.

6.7.2.2 Utilización de librería para generación de gráficos JFreechart

JFreechart [Dan13] es una librería de código abierto disponible para Java que permite a los usuarios generar fácilmente gráficos y diagramas. Para integrar esta librería en el código de la aplicación AAS11 se han llevado a cabo los siguientes pasos:

- Implementación de la clase que representa un gráfico dentro de la aplicación AAS11: en la siguiente figura se puede observar la clase GraficoTO en la que se han definido los atributos IdGrafico (identificador del tipo de gráfico a utilizar) y nombGrafico (nombre del gráfico a utilizar) además del conjunto de métodos que permiten el acceso a los mismos.

```
package org.bastanchu.pfc.aas11.to;
public class GraficoTO {
	int IdGrafico;
	String nombGrafico;

	public int getIdGrafico() {
		return IdGrafico;
	}

	public void setIdGrafico(int idGrafico) {
		IdGrafico = idGrafico;
	}

	public String getNombGrafico() {
		return nombGrafico;
	}

	public void setNombGrafico(String nombGrafico) {
		this.nombGrafico = nombGrafico;
	}

}
```

Figura 95.Clase GraficoTO.

- Implementación del servicio que permite la construcción de gráficos en la aplicación AAS11: el primer paso que se ha llevado a cabo ha sido la definición de la interfaz asociado a dicho servicio y que se muestra en la siguiente figura:

```
package org.bastanchu.pfc.aas11.service;

import java.util.List;

import org.bastanchu.pfc.aas11.to.GraficoTO;
import org.bastanchu.pfc.aas11.to.ResultadoConsultaGenericaTO;

public interface GraficoService {
    public List<GraficoTO> obtenerTodosLosGraficos();
    public List<GraficoTO> obtenerUnGrafico(int idGrafico);

    public byte[] obtenerDiagramaDeBarras(ResultadoConsultaGenericaTO
       resultado);

    public byte[] obtenerDiagramaDeTarta(ResultadoConsultaGenericaTO
       resultado);

}
```

Figura 96. Interfaz GraficoService.

En esta interfaz se han declarado los métodos: obtenerTodosLosGraficos, obtenerUnGrafico, obtenerDiagramaDeBarras y obtenerDiagramaDeTarta. La implementación de estos métodos se ha llevado a cabo en la clase GraficoServiceImpl y se muestra en las siguientes figuras:

```
package org.bastanchu.pfc.aas11.service.impl;
import java.awt.Font;

@Service
@Transactional(propagation = Propagation.REQUIRED)
public class GraficoServiceImpl implements GraficoService {
    private GraficoDao listaGraficoDao;
    public GraficoDao getListaGraficoDao() {
            return listaGraficoDao;
    }
    @Inject
    public void setListaGraficoDao(GraficoDao listaGraficoDao) {
            this.listaGraficoDao = listaGraficoDao;
    }
    public List<GraficoTO> obtenerTodosLosGraficos() {
            return this.getListaGraficoDao().obtenerTodos();
    }
    public List<GraficoTO> obtenerUnGrafico(int idGrafico) {
            return this.getListaGraficoDao().obtenerUnGrafico(idGrafico);
    }

    @Override
    public byte[] obtenerDiagramaDeBarras(ResultadoConsultaGenericaTO resultado) {
            // TODO Auto-generated method stub
            List<Map<Integer, Object>> datos = resultado.getDatos();
            DefaultCategoryDataset dataset = new DefaultCategoryDataset();
            for (Map<Integer, Object> unDato : datos) {
                    try {
                            double valor = Double.parseDouble(unDato.get(1).toString());
                            dataset.setValue(valor, "valor", unDato.get(0).toString());
                    } catch (NumberFormatException e) {
                            throw new RuntimeException("Dato no valido para diagrama de
                                barras");
                    }
            }
            JFreeChart diagramaDeBarras = ChartFactory.createBarChart("Diagrama de
                barras", "Datos" , "valor", dataset, PlotOrientation.VERTICAL, false, true,
                false);
            ByteArrayOutputStream out = new ByteArrayOutputStream();
            try {
                    ChartUtilities.writeChartAsJPEG(out, diagramaDeBarras, 500, 300);
            } catch (IOException e) {
                    throw new RuntimeException("Error al genera imagen");
            }
            byte[] outputBytes = out.toByteArray();
            return outputBytes;
    }
```

Figura 97. Clase GraficoServiceImpl parte 1.

```
@Override
public byte[] obtenerDiagramaDeTarta(ResultadoConsultaGenericaTO resultado) {
        List<Map<Integer, Object>> datos = resultado.getDatos();
        DefaultPieDataset dataset = new DefaultPieDataset();
        for (Map<Integer, Object> unDato : datos) {
                try {
                        double valor = Double.parseDouble(unDato.get(1).toString());
                        dataset.setValue(unDato.get(0).toString(), valor);
                } catch (NumberFormatException e) {
                        throw new RuntimeException("Dato no valido para diagrama de
                        tarta");
                }
        }
        JFreeChart diagramaDeTarta = ChartFactory.createPieChart("Diagrama de
        tarta", dataset, true, false, false);
        PiePlot plot = (PiePlot) diagramaDeTarta.getPlot();
        plot.setLabelFont(new Font("SansSerif", Font.PLAIN, 12));
        plot.setNoDataMessage("No data available");
        plot.setCircular(false);
        plot.setLabelGap(0.02);
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        try {
                ChartUtilities.writeChartAsJPEG(out, diagramaDeTarta, 500, 300);
        } catch (IOException e) {
                throw new RuntimeException("Error al genera imagen");
        }
        byte[] outputBytes = out.toByteArray();
        return outputBytes;
}


}
```

Figura 98. Clase GraficoServiceImpl parte 2.

En la clase GraficoServiceImpl destacan los métodos obtenerDiagramaDeBarras y obtenerDiagramaDeTarta que devuelven un array de bytes que contiene una imagen JPEG con el dibujo del diagrama correspondiente. La primera acción que se realiza en estos métodos es componer el conjunto de datos a representar en el diagrama. El siguiente paso será establecer el conjunto de parámetros asociados al diagrama seleccionado (barras o tarta). La última acción, será la de conversión del diagrama a un formato de imagen (JPEG) para ser devuelto como un array de bytes.

- Composición de gráficos en el método mostrarGraficoTuning: el paso final en la utilización de gráficos, será invocar a la generación de imágenes desde el Controller correspondiente para visualizarlas, en última instancia, por la pantalla. Esta acción es realizada por el método mostrarGraficoTunig perteneciente a la clase NuevoInformeController que se muestra en la siguiente figura:

```
@RequestMapping("/mostrarGraficoTuning")
public ModelAndView mostrarGraficoTuning(Integer idCuestion,
HttpServletResponse response) {
        try {
                CuestionTuningTO cuestionTuning =
                this.getCuestionTuningService().consultarCuestionTuning(idCuestion);
                ResultadoConsultaGenericaTO resultadoConsulta =
                 this.getConsultaGenericaService().realizarConsultaGenerica(
                cuestionTuning.getSelCuestion());
                byte[] diagramaJPEG = null;
                if (cuestionTuning.getIdGrafico() == 1) {
                        diagramaJPEG=this.getGraficoService().
                        obtenerDiagramaDeBarras(resultadoConsulta);
                }
                else {
                        diagramaJPEG =this.getGraficoService().
                        obtenerDiagramaDeTarta(resultadoConsulta);
                }
                response.setContentType("image/jpeg");
                response.getOutputStream().write(diagramaJPEG);
                return null;
        } catch (IOException e) {
                throw new RuntimeException("Error al escribir imagen");
        }
}
```

Figura 99. Método mostrarGraficoTuning.

En las siguientes figuras se pueden observar ejemplos de gráficos generados a partir de la librería gráfica JFreeChart y utilizados por la aplicación AAS11:

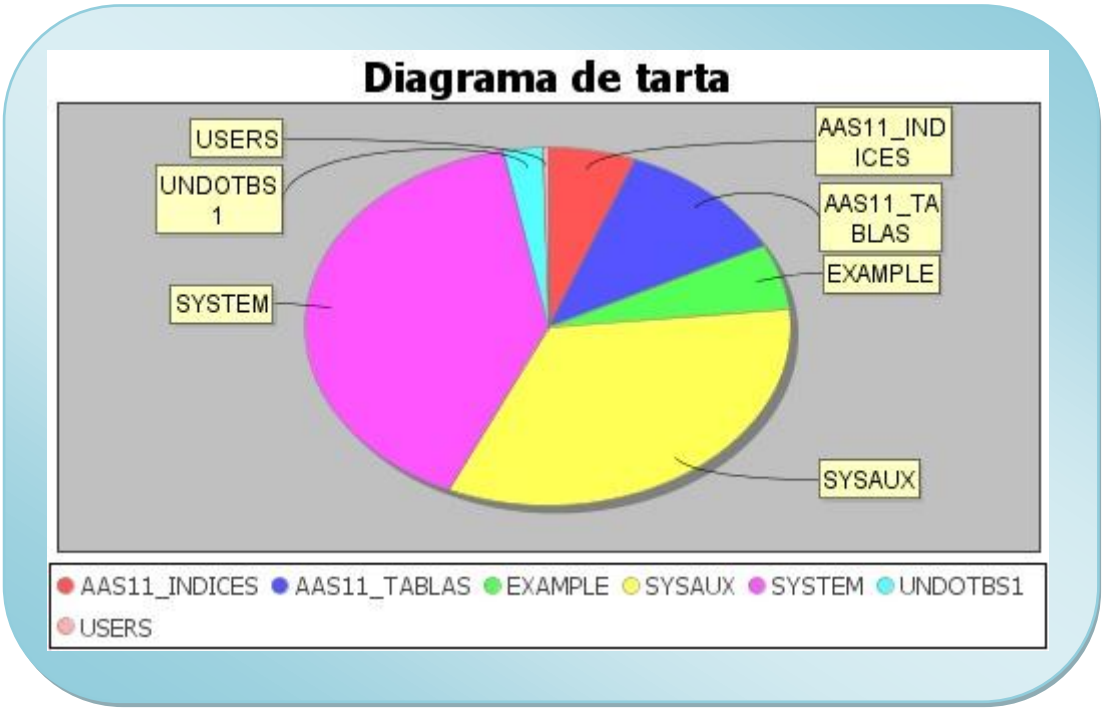


Figura 100. Diagrama de tarta.Tamaño actual de Tablespaces expresado en MB .

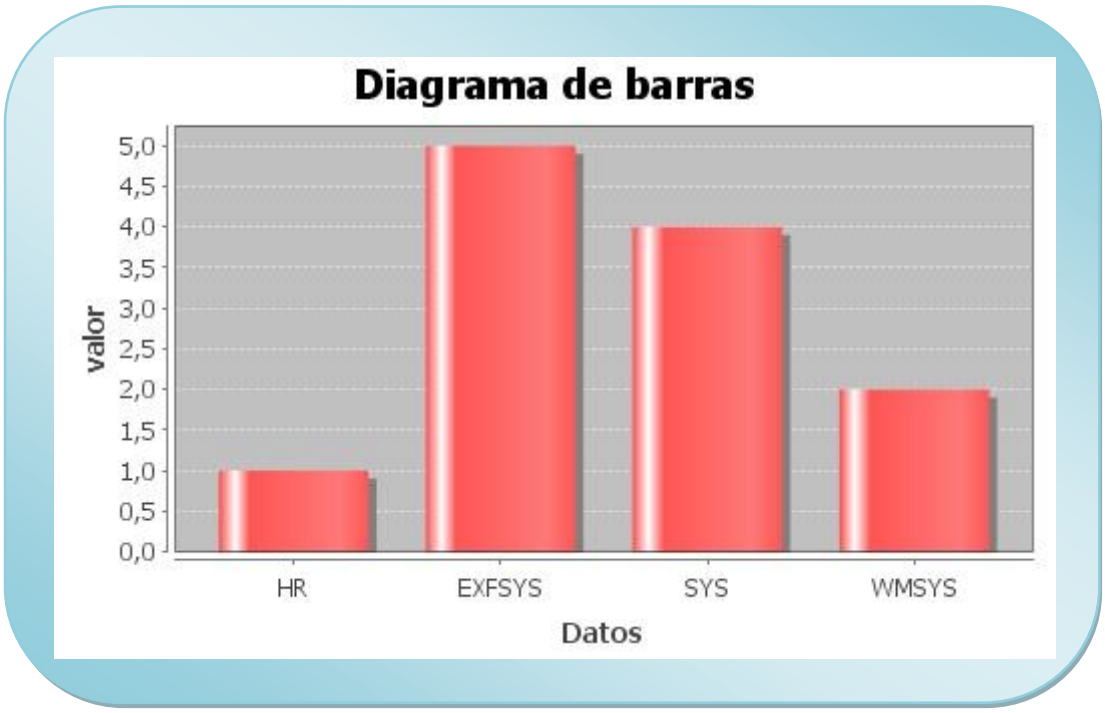


Figura 101. Diagrama de barras.Triggers deshabilitados clasificados por propietario.

6.7.2.3 Almacenamiento de documento Word utilizando un BLOB

Un BLOB (Binary Large OBject) [OraDocs] es una cadena binaria de longitud variable que puede se contener hasta 2 Gigabytes. Al igual que otros tipos binarios, las cadenas BLOB no están asociadas a un tipo de codificación específica. Este tipo de datos ha sido utilizado para almacenar los documentos Word obtenidos como resultado de la realización de test de auditoría y de tuning para permitir su posterior recuperación. Los elementos relacionados con la utilización de este tipo de datos particular aparecen descritos a continuación:

- Implementación de la clase que representa un informe en la aplicación AAS11: la clase InformeTO representa los informes utilizados por la aplicación AAS11. Los atributos asociados a un informe son:

    - idInforme: identificador del informe de auditoría/tuning.
    - desinforme: descripción asociada al informe de auditoría/tuning.
    - fechaInforme: fecha establecida para el informe de auditoría/tuning.
    - idUsuario: identificador de usuario que ha elaborado el informe de auditoría/tuning.
    - contenidoInforme: documento Word que constituye el informe de auditoría/tuning elaborado por un usuario.

En la siguiente figura aparece la clase InformeTO:

```
package org.bastanchu.pfc.aas11.to;

import java.util.Date;

public class InformeTO {

	private String idInforme;
	private String desInforme;
	private Date fechaInforme;
	private String idUsuario;
	private byte[] contenidoInforme;

	public String getIdInforme() {
		return idInforme;
	}
	public void setIdInforme(String idInforme) {
		this.idInforme = idInforme;
	}
	public String getDesInforme() {
		return desInforme;
	}
	public void setDesInforme(String desInforme) {
		this.desInforme = desInforme;
	}
	public Date getFechaInforme() {
		return fechaInforme;
	}
	public void setFechaInforme(Date fechaInforme) {
		this.fechaInforme = fechaInforme;
	}
	public String getIdUsuario() {
		return idUsuario;
	}
	public void setIdUsuario(String idUsuario) {
		this.idUsuario = idUsuario;
	}

	public byte[] getContenidoInforme() {
		return contenidoInforme;
	}
	public void setContenidoInforme(byte[] contenidoInforme) {
		this.contenidoInforme = contenidoInforme;
	}
}
```

Figura 102. Clase InformeTO.

- Implementación de la clase DAO InformeDaoImpl que permite llevar a cabo la inserción en la base de datos: esta clase se utilizará para llevar a cabo la inserción de un nuevo informe en la base de datos para permitir su posterior recuperación.

En la siguiente figura aparece el método insertar perteneciente a la clase InformeDaoImpl utilizado para realizar esta acción:

```
@Override
public void insertar(final InformeTO entidad) {
        JdbcTemplate jdbcTemplate = this.getJdbcTemplate();
        String consulta = "insert into INFORMES (IDINFORME, DESINFORME,
                          FECHAINFORME, IDUSUARIO, CONTENIDOINFORME)"
                          + " values (?,?,?,?,?)";
        jdbcTemplate.execute(consulta, new PreparedStatementCallback<InformeTO>()
        {
                @Override
                public InformeTO doInPreparedStatement(PreparedStatement
                    statement) throws SQLException, DataAccessException {
                    statement.setString(1, entidad.getIdInforme());
                    statement.setString(2, entidad.getDesInforme());
                    statement.setTimestamp(3, new
                        Timestamp(entidad.getFechaInforme().getTime()));
                    statement.setString(4, entidad.getIdUsuario());
                    InputStream stream = new
                        ByteArrayInputStream(entidad.getContenidoInforme());
                    statement.setBlob(5, stream);
                    return entidad;
                }
        });
    }
```

Figura 103. Método insertar perteneciente a la clase InformeDaoImpl.

- Implementación del método guardarInformeWord en la clase InformeWordServiceImpl que representa el servicio asociado a la funcionalidad de tratamiento de informes:

```
@Override
@Transactional
public void guardarInformeWord(InformeTO informe) {
        this.getInformeDao().insertar(informe);
}
```

Figura 104. Método guardarInformeWord perteneciente a la clase InformeWordServiceImpl.

- Implementación del método guardarInforme perteneciente a la clase NuevoInformeController utilizado para proceder a componer el informe de auditoría/tuning y llevar a cabo su almacenamiento en la base de datos:

```
private void guardarInforme(InformeForm form, XWPFDocument document, HttpSession
sesion)
        throws IOException {
        InformeTO informeTO = new InformeTO();
        informeTO.setIdInforme(form.getIdInforme());
        informeTO.setDesInforme(form.getDesInforme());
        informeTO.setFechaInforme(form.getFechaInforme());
        UsuarioAutenticadoSesionTO usuarioSessionTO =
        (UsuarioAutenticadoSesionTO) sesion.getAttribute("datosUsuarioSesion");
        informeTO.setIdUsuario(usuarioSessionTO.getUsuario());
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        document.write(stream);
        informeTO.setContenidoInforme(stream.toByteArray());
        this.getInformeWordService().guardarInformeWord(informeTO);
}
```

Figura 105. Método guardarInforme perteneciente a la clase NuevoInformeController.

6.7.3 Reunión de revisión y retrospectiva del Sprint 4

En esta reunión se realiza un análisis del Sprint llevado a cabo con el objetivo de analizar los resultados finales de la implementación de la aplicación AAS11 y de evaluar las dificultades encontradas en la implementación de los últimos requisitos. Con este Sprint finaliza la implementación de requisitos y tras su consecución se produce el hito de finalización del proyecto.

En la siguiente figura podemos ver el diagrama Burn Down asociado al último Sprint llevado a cabo:

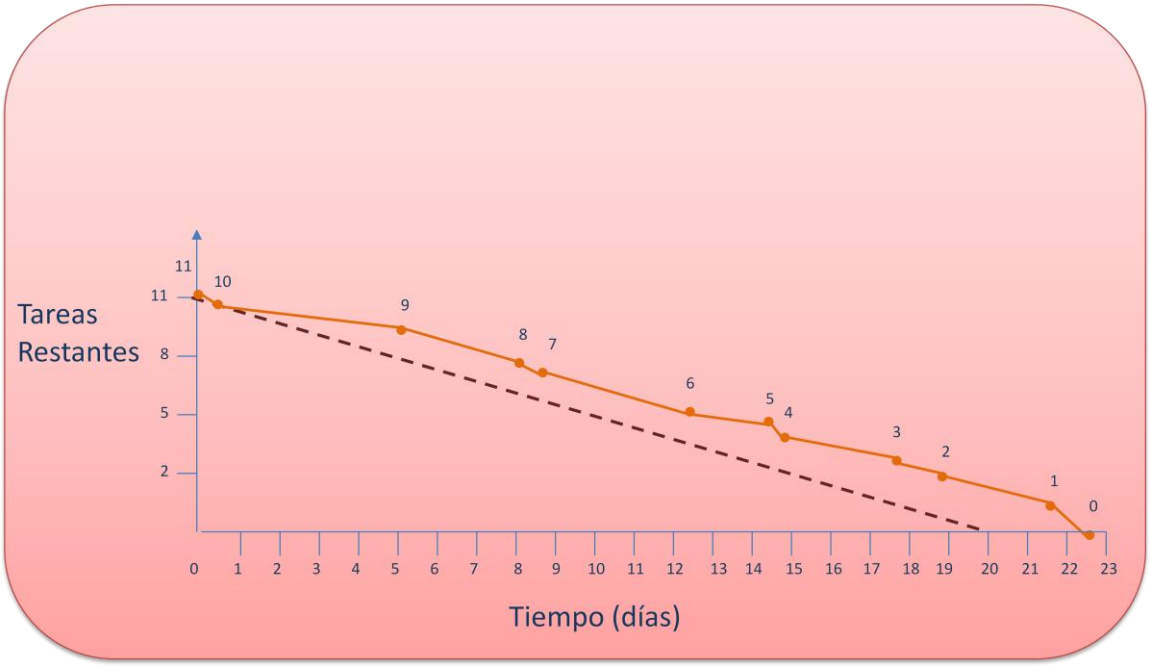


Figura 106.Diagrama Burn-Down asociado al Sprint 4.

En este diagrama se puede observar cómo se ha producido una desviación significativa con respecto a la planificación inicialmente establecida. Los puntos en los que se ha encontrado mayor dificultad coinciden con aquellos en los que se ha necesitado mayor cantidad de tiempo para alcanzar su resolución. En particular, las tareas que han exigido mayor dedicación han sido la construcción de la opción que permite a los usuarios llevar a cabo test personalizados y la opción que permite realizar la exportación de los resultados a formato de documento Word (seleccionado por su amplia difusión).

Capítulo 7

Presupuesto

7.1 Introducción

En este capítulo se detallará el presupuesto elaborado para afrontar la implementación de la aplicación Automatic Audit System for Oracle 11g (AAS11). Esta aplicación se utilizará como herramienta disponible sobre entornos en los que se encuentre instalado el sgbdr Oracle 11g. A través de esta herramienta, se podrá llevar a cabo una evaluación del estado de un determinado sistema, planteando una serie de cuestiones de auditoría y de tuning, que deben ser validadas por parte del usuario. Una vez respondidas la totalidad de cuestiones, se podrá generar un informe de auditoría que constituirá un valioso instrumento, a disposición del auditor, que puede utilizarse como referencia en la elaboración del informe de auditoría final.

Los puntos que se abordarán en este capítulo se exponen a continuación:

- División en fases y subfases del proyecto.
- Diagrama de Gantt.
- Resumen de costes.

7.2 División en fases y subfases

A continuación se presenta el detalle de cada una de las fases y subfases que se han abordado para llevar a cabo la implementación de la aplicación AAS11.

7.2.1 Definición de requisitos

Como fase previa al desarrollo de la aplicación AAS11 se ha procedido a analizar la documentación necesaria para establecer una definición de requisitos completa. Con este objetivo, la documentación analizada ha sido la siguiente:

- Análisis de la LOPD y Real Decreto 1720/2007: en el análisis realizado, se ha establecido una comparativa entre los preceptos que establecen y los mecanismos de los que dispone el sgbdr de Oracle para adaptarlos.

- El estándar internacional ISO/IEC 27002: en este caso, se ha llevado a cabo una selección de aquellos puntos que tienen especial relación con los sgbdr. A partir de esta selección, se lleva a cabo una comparativa entre estos puntos y los mecanismos de los que dispone Oracle para satisfacer los requerimientos establecidos.

- Metodología ISACA (COBIT 4.1): con respecto a la documentación asociada al COBIT, se ha procedido a realizar una exposición detallada de los objetivos de control que tienen particular relación con las bases de datos.

La siguiente fase, después de llevar a cabo el análisis de la documentación que se ha tomado como referencia, ha sido seleccionar entre diversas opciones[101], una lista de comprobación que englobe todos los aspectos de la documentación que han sido considerados cómo relevantes. La lista de comprobación que se ha seleccionado finalmente ha sido el CIS Benchmark para Oracle.

En la siguiente figura se representa la documentación que se ha tomado como base y la lista de comprobación que se ha seleccionado (en esta lista confluyen los elementos susceptibles de análisis en un proceso de auditoría).

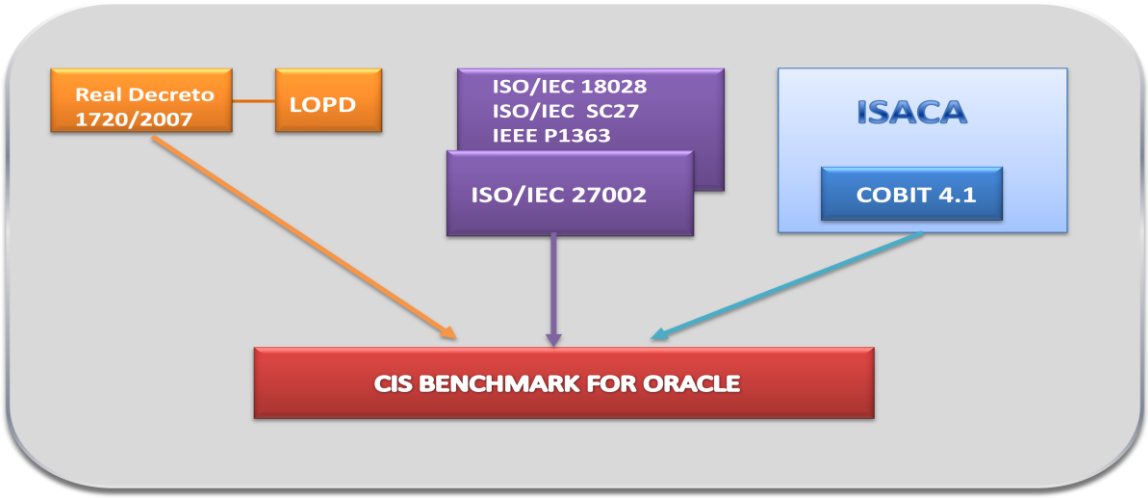


Figura 107.Relación entre la documentación analizada y la lista de comprobación.

[101] Otra de las opciones contempladas ha sido la lista de comprobación Database STIG (Security Technical Implementation Guides) publicada por DISA (Defense Information Systems Agency) y utilizada por el DoD (Department of Defense) disponible http://iase.disa.mil/stigs/checklist/index.html.

La siguiente tabla expone cada una de las fases-subfases llevadas a cabo junto con su duración en días:

Fase	Subfase	Id Subfase[102]	Duración (días)
Análisis de Documentación	Análisis de LOPD y Real Decreto 1720/2007	4	4
	Análisis de ISO/IEC 27002	5	5
	Análisis del COBIT	6	4
Selección de lista de comprobación	CIS Benchmark para Oracle	9	3

Tabla 57.Fases y subfases asociadas a la definición de requisitos.

7.2.2 Sprint 1

En el Sprint 1 se desarrollan tareas asociadas a una serie de requisitos relacionados con la implementación de la interfaz web, gestión de usuarios e implementación de la opción que permite la consulta de los perfiles dados de alta en la aplicación:

Fase	Subfase	Id Subfase	Duración (días)
Implementación del requisito PB-0-001	Selección de la arquitectura (framework, servidor de aplicaciones, herramientas de control y software subyacente).	12	3
	Configuración y parametrización de aplicación.	13	1
	Diseño de la capa de presentación de la aplicación.	14	1
	Elaboración de un prototipo para realizar su validación.	15	1
Implementación del requisito PB-0-002	Planteamiento del requisito utilizando un diagrama de casos de uso.	17	0,25
	Generación de esquema de base de datos.	18	1
	Planteamiento inicial del modelo de datos necesario para llevar a cabo la gestión de usuarios.	19	0,5
	Implementación de la parte del modelo de datos que permite dotar de esta funcionalidad.	20	0,25
	Carga de datos en base de datos.	21	0,25
	Construcción de procedimiento de Login para permitir la entrada de usuarios en la aplicación.	22	2
	Prueba de procedimiento de Login.	23	0,5
	Construcción de opción de “Alta de Usuarios” en la aplicación.	24	2
	Prueba de opción de “Alta de Usuarios”.	25	0,5
	Construcción de opción de “Modificación de Usuarios” en la aplicación.	26	2
	Prueba de opción de “Modificación de Usuarios”.	27	0,5
	Construcción de opción de “Baja de Usuarios” en la aplicación.	28	2
	Prueba de opción de “Baja de Usuarios”.	29	0,5

Tabla 58.Fases y subfases asociadas al Sprint 1.Parte I.

[102] Identificador de la subfase empleado en el diagrama de Gantt.

Fase	Subfase	Id Subfase	Duración (días)
Implementación del requisito PB-0-003	Planteamiento del requisito utilizando un diagrama de casos de uso.	31	0,25
	Introducción en el modelo de datos de la gestión de perfiles de usuario.	32	0,5
	Carga de datos en la base de datos.	33	0,25
	Construcción de la opción “Consulta de perfiles/menús/opciones”.	34	1
	Prueba de la opción de consulta de “Consulta de perfiles/menús/opciones”.	35	0,5

Tabla 59.Fases y subfases asociadas al Sprint 1.Parte II.

7.2.3 Sprint 2

En este Sprint se implementan las funcionalidades relativas a la gestión de cuestiones de auditoría y a la gestión de cuestiones de tuning. Estas opciones permitirán la administración de estos tipos de cuestiones en la aplicación AAS11.

Fase	Subfase	Id Subfase	Duración (días)
Implementación del requisito PB-0-004 y PB-0-008	Inclusión en el modelo de datos de las cuestiones de auditoría y tuning.	39	0,5
	Elaboración de clases TO y FORM utilizadas para manipular las cuestiones de auditoría y tuning.	40	3
Implementación del requisito PB-0-005 y PB-0-009	Planteamiento del requisito utilizando un diagrama de casos de uso.	42	0,25
	Carga de datos en base de datos para permitir la realización de pruebas sobre cada una de las opciones.	43	3
	Construcción de las opciones de “Alta de cuestiones de auditoría” y “Alta de cuestiones de tuning”.	44	3
	Prueba de la opciones de “Alta de cuestiones de auditoría” y “Alta de cuestiones de tuning”.	45	1
	Construcción de las opciones de “Modificación de cuestiones de auditoría” y “Modificación de cuestiones de tuning” en la aplicación.	46	4
	Prueba de la opciones de “Modificación de cuestiones de auditoría” y “Modificación de cuestiones de tuning” en la aplicación.	47	1
	Construcción de opciones de “Baja de cuestiones de auditoría” y “Baja de cuestiones de tuning”.	48	4
	Prueba de opciones de “Baja de cuestiones de auditoría” y “Baja de cuestiones de tuning”.	49	1

Tabla 60.Fases y subfases asociadas al Sprint 2.

7.2.4 Sprint 3

En el Sprint 3 se procede a desarrollar el conjunto de requisitos relativos a la gestión de parte de la información almacenada en la aplicación. Estos requisitos incluyen el conjunto de opciones que se utilizarán para clasificar las cuestiones de auditoría y las cuestiones de tuning en secciones y apartados respectivamente. Adicionalmente en este Sprint se implementarán las opciones que permitirán particularizar la selección de cuestiones de auditoría y de las cuestiones de tuning para cada usuario.

En el Sprint 3 se incorpora la utilización de Hibernate Validator con el objetivo de controlar la información introducida a través de los formularios presentados en pantalla. Esta herramienta permitirá mantener la coherencia de la información almacenada en la base de datos.

Fase	Subfase	Id Subfase	Duración (días)
Implementación del requisito PB-0-006 y PB-0-010	Planteamiento del requisito utilizando un diagrama de casos de uso.	53	0,25
	Introducción en el modelo de datos de la clasificación de cuestiones de auditoría y la clasificación de cuestiones de tuning.	54	0,5
	Carga de datos en la base de datos para la realización de pruebas posteriores.	55	0,5
	Construcción de las opciones de “Alta de secciones” y “Alta de apartados”.	56	2
	Prueba de la opciones de “Alta de secciones” y “Alta de apartados”.	57	1
	Construcción de las opciones de “Modificación de secciones” y “Modificación de apartados” en la aplicación.	58	3
	Prueba de la opciones de “Modificación de cuestiones de secciones” y “Modificación de apartados” en la aplicación.	59	1
	Construcción de opciones de “Baja de secciones” y “Baja de apartados”.	60	3
	Prueba de opciones de “Baja de secciones” y “Baja de apartados”.	61	1
	Utilización de Hibernate Validator en los formularios ya implementados.	62	2
Implementación del requisito PB-0-007 y PB-0-011	Planteamiento del requisito utilizando un diagrama de casos de uso.	64	0,25
	Introducción en el modelo de datos de la selección personalizada de cuestiones de auditoría y tuning.	65	0,5
	Construcción de las opciones de “Selección de cuestiones de auditoría” y “Selección de cuestiones de tuning”.	66	3
	Prueba de opciones de “Selección de cuestiones de auditoría” y “Selección de cuestiones de tuning”.	67	2

Tabla 61.Fases y subfases asociadas al Sprint 3.

7.2.5 Sprint 4

En el Sprint 4 se afrontan los últimos requisitos definidos en la aplicación. Estos requisitos hacen referencia a la funcionalidad relativa a la elaboración de los test por parte de los usuarios. Adicionalmente en estos requisitos se establece la obligatoriedad de permitir la generación de los resultados de los test realizados por cada usuario en formato Word.

Fase	Subfase	Id Subfase	Duración (días)
Implementación del requisito PB-0-012	Planteamiento del requisito utilizando un diagrama de casos de uso.	71	0,25
	Construcción de la opción que permite la realización de los test a los usuarios.	72	5
	Prueba de las opción que permite la realización de test a los usuarios.	73	3
Implementación del requisito PB-0-014	Planteamiento del requisito utilizando un diagrama de casos de uso.	75	0,25
	Desarrollo e integración de la funcionalidad que permite exportar el resultado de los Test a formato Word.	76	4
	Prueba de las opción que permite la realización de test a los usuarios.	77	2
Implementación del requisito PB-0-013	Planteamiento del requisito utilizando un diagrama de casos de uso.	79	0,25
	Construcción de la opción que permite la consulta de los Test realizados a cada usuario.	80	3
	Prueba de las opción que permite la consulta de test a cada usuario.	81	1
	Construcción de la opción que permite la eliminación de los Test realizados a cada usuario.	82	3
	Prueba de las opción que permite la eliminación de test a cada usuario.	83	1

Tabla 62.Fases y subfases asociadas al Sprint 4.

7.2.6 Ayuda contextual de la aplicación AAS11

Después de llevar a cabo la implementación de la herramienta AAS11 se procede a completar el contenido de la ayuda contextual de la aplicación.

Fase	Subfase	Id Subfase	Duración (días)
Ayuda Contextual	Completar contenido de la ayuda contextual.	86	6

Tabla 63. Fases y subfases asociadas a la ayuda contextual.

7.3 Diagrama de Gantt

El diagrama de Gantt constituye una herramienta que permite representar el cronograma de tareas llevadas a cabo en la implementación del proyecto AAS11.

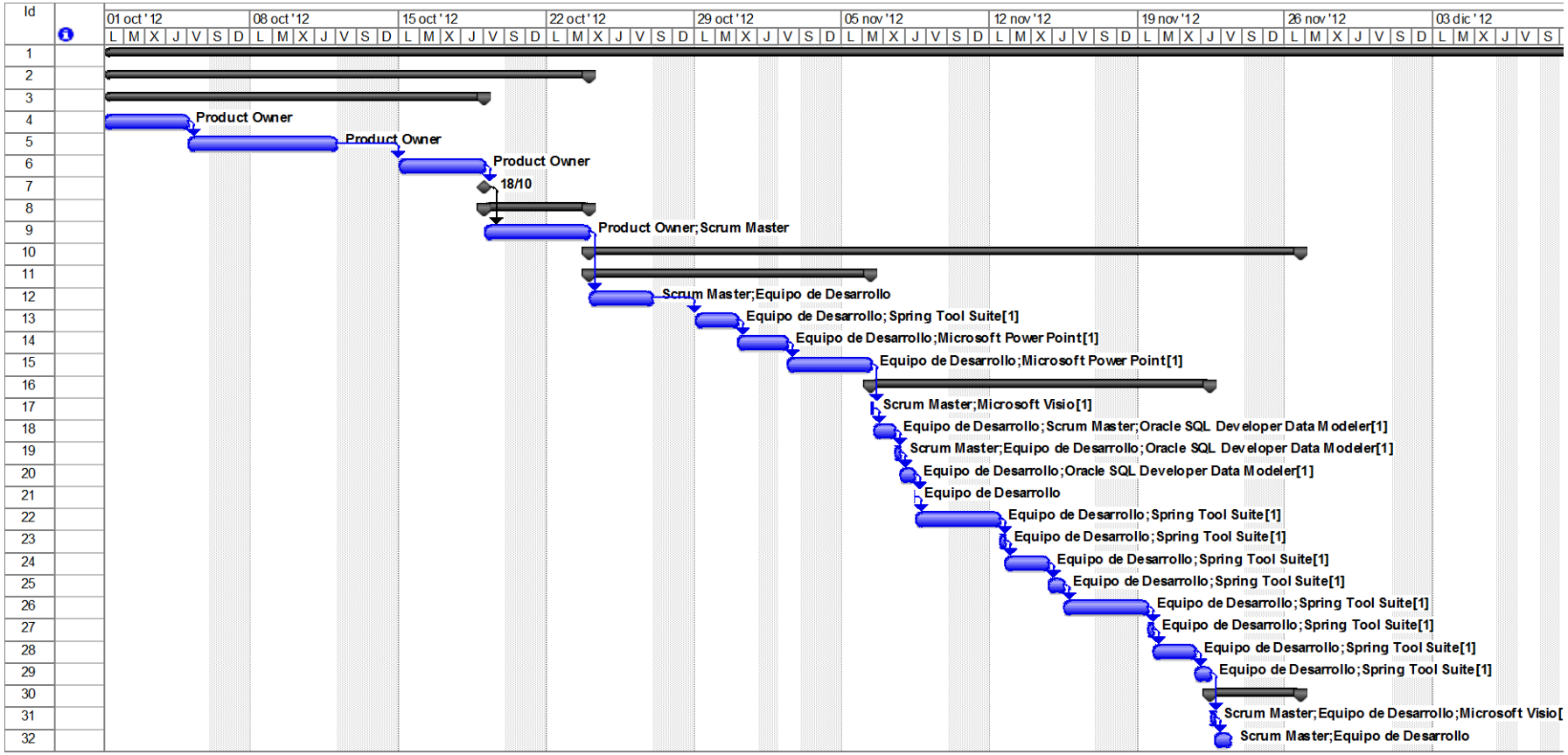


Figura 108.Diagrama de Gantt asociado al proyecto AAS11, parte 1/3.

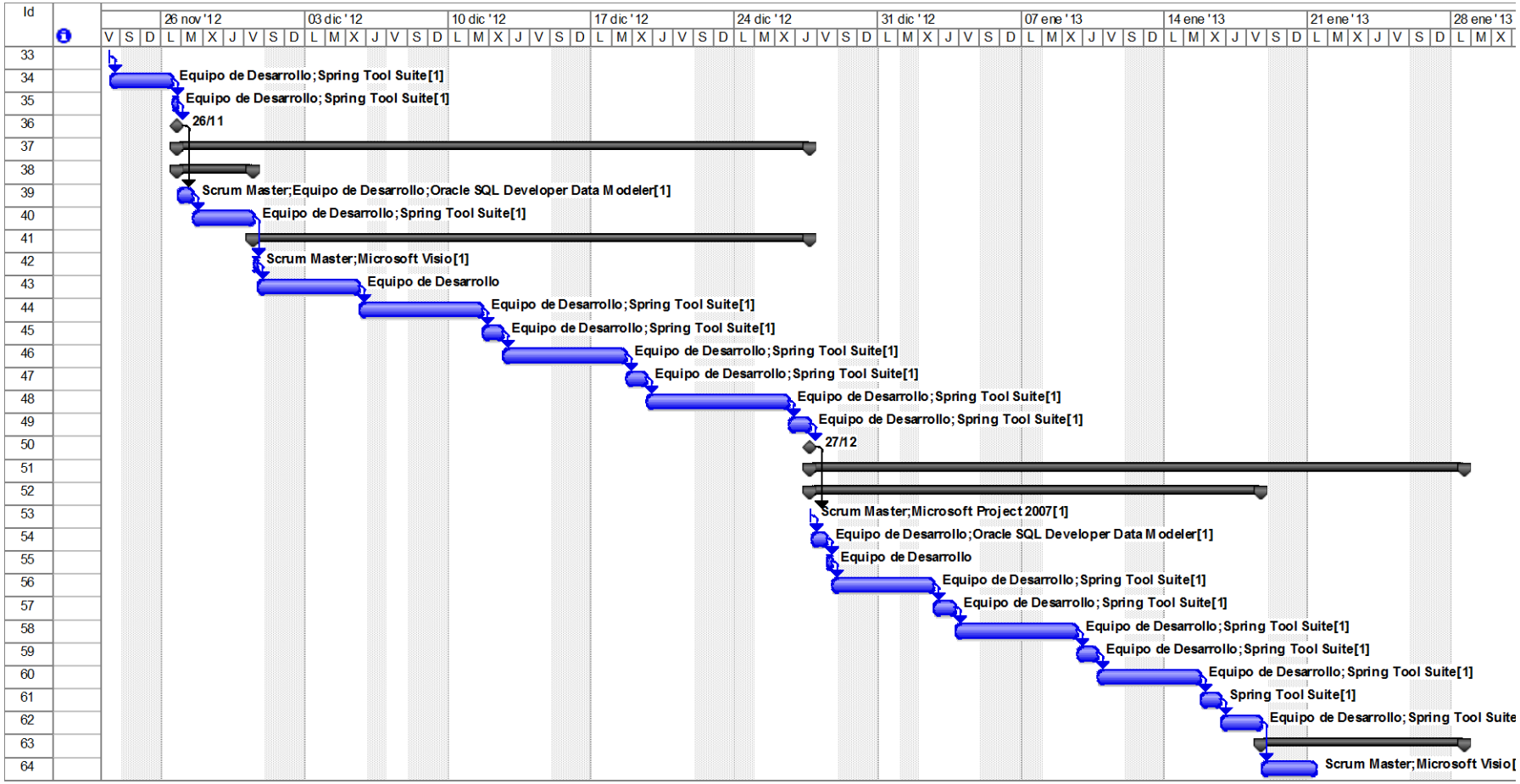


Figura 109.Diagrama de Gantt asociado al proyecto AAS11, parte 2/3.

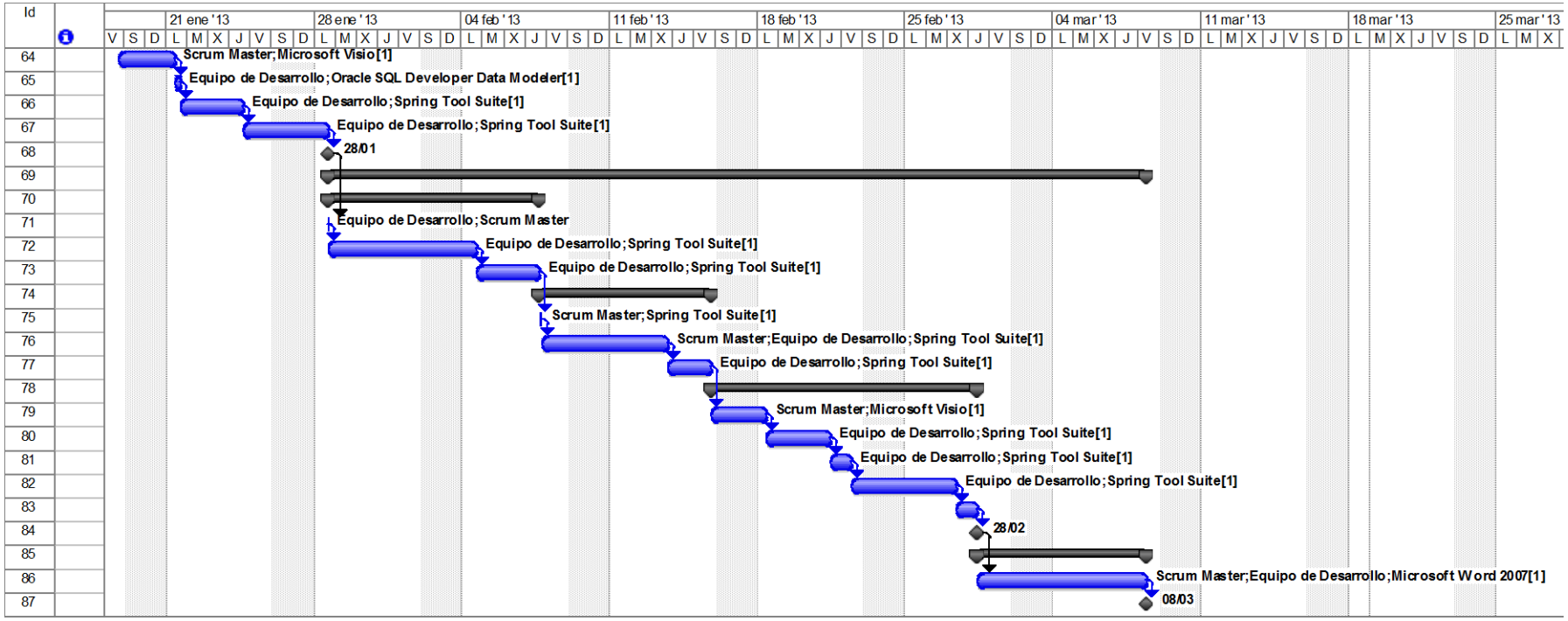


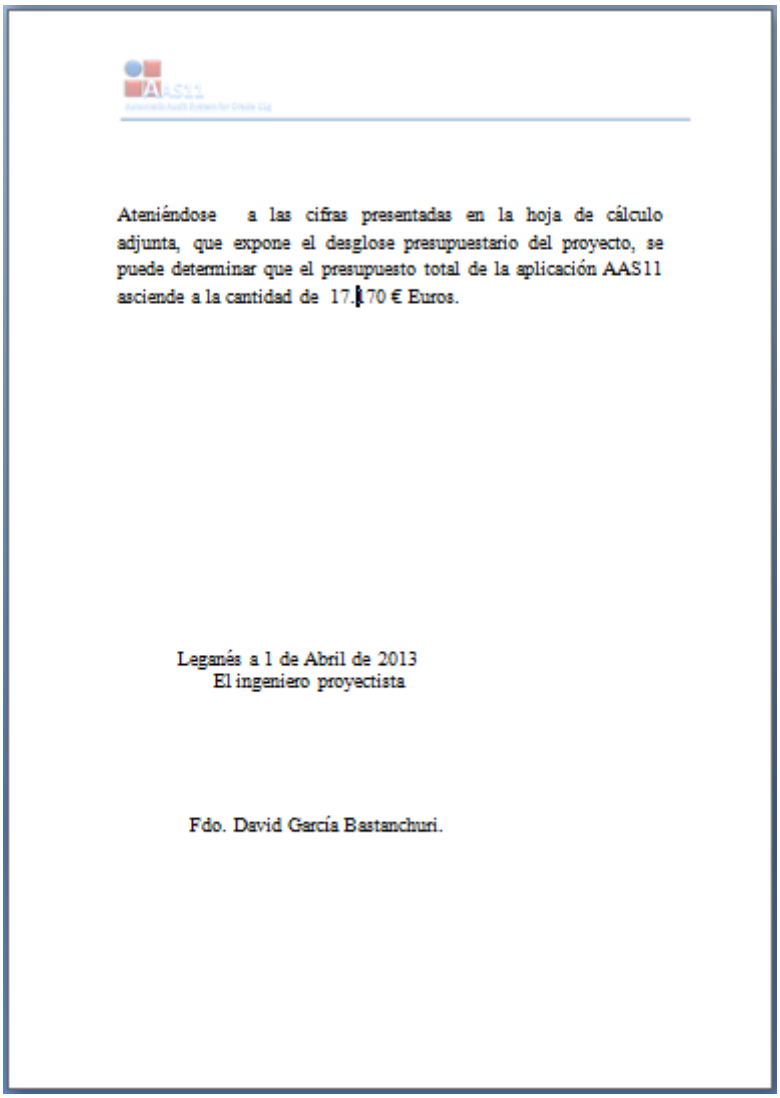
Figura 110.Diagrama de Gantt asociado al proyecto AAS11, parte 3/3.

En el diagrama de Gantt se puede apreciar que la duración de la implementación del proyecto comprende del 1/10/2012 al 8/3/2013. Por cada tarea, aparecen los recursos personales y de software implicados. Los hitos que aparecen reflejados en el diagrama incluyen: la elaboración del product backlog, las reuniones de retrospectiva asociadas a cada Sprint y el hito que identifica el fin del proyecto.

7.4 Resumen de costes

En este punto se presenta el resumen de costes asociado a la implementación de la aplicación AAS11. El calendario laboral establecido, implica la realización de jornadas de 5 horas. El tiempo real invertido en el desarrollo del proyecto ha sido distribuido de forma más irregular, aún así, el calendario expuesto en las figuras anteriores refleja de forma fidedigna, en media, el periodo de desarrollo de la aplicación.

La siguiente figura muestra el modelo de documento empleado en la presentación del presupuesto asociado al desarrollo de la aplicación AAS11:



Ateniéndose a las cifras presentadas en la hoja de cálculo adjunta, que expone el desglose presupuestario del proyecto, se puede determinar que el presupuesto total de la aplicación AAS11 asciende a la cantidad de 17.170 € Euros.

Leganés a 1 de Abril de 2013
El ingeniero proyectista

Fdo. David García Bastanchuri.

Figura 111.Hoja resumen del presupuesto asociado al proyecto AAS11.

A continuación se expone la hoja de cálculo utilizada para llevar a cabo la valoración económica del de desarrollo de la aplicación AAS11:

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor: David García Bastanchuri

2.- Departamento: Ingeniería Informática

3. Descripción del Proyecto:

Título	AAS11 Automatic Audit System for Oracle 11g
Duración (meses)	5,25
Tasa de costes Indirectos:	20%

4.- Presupuesto total del Proyecto (valores en Euros):

Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
		Product Owner	0,5	4.000,00	2.000,00	
		Scrum Master	1,75	3.400,00	5.950,00	
		Equipo de desarrollo	3	2.100,00	6.300,00	
					0,00	
					0,00	
		Hombres mes	5,25	Total	14.250,00	

[a] 1 Hombre mes = 100 horas.

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Intel Core II duo 4 Gb Memory 64	750,00	70	5	60	45,94
Microsoft Office Professional Edi	300,00	50	5	60	12,50
Glassfish Server Open Source Edi	0,00	100	5	60	0,00
Spring tool suite 3.1.0 RELEASE	0,00	100	5	60	0,00
Google Chrome 25.0.1364.172	0,00	80	5	60	0,00
Oracle 11 g EE para desarrollo	0,00	50	5	60	0,00
				Total	58,44

[d] Fórmula de cálculo de la Amortización.

$$\frac{A}{B} x C x D$$

A – nº de meses desde la fecha de facturación en que el equipo es utilizado
B – período de depreciación (60 meses)
C – coste del equipo (sin IVA)
D – % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
	Total	0,00

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas,

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	14.250
Amortización	58
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	2.862
Total	17.170

Figura 112.Fichero Excel utilizado en el cálculo del presupuesto asociado al proyecto.

Como puede observarse en la figura anterior, la hoja de cálculo utiliza la medida hombre/mes[103] para establecer el coste del proyecto AAS11. Esta medida debe considerarse a efectos únicos de determinar el coste de un proyecto, pero en ningún caso debe ser considerada como una medida de alcance del proyecto.

El coste total del proyecto, al considerar una tasa de costes indirectos del 20%, asciende a la cantidad de **17170 Euros**.

[103] Algunos autores de reconocido prestigio como Frederick Brooks consideran en parte de su biografía [Bro75] algunos errores frecuentes en la gestión de proyectos software, y en especial el relacionado con la medida hombre/mes. Una de sus observaciones, es la denominada Ley de Brooks que establece que: Añadir más recursos a un proyecto con retraso podría demorarlo aún más.

Capítulo 8

Conclusiones

8.1 Introducción

En este capítulo se realizará una exposición de las conclusiones extraídas en el proyecto, partiendo de la base de los objetivos planteados al inicio. Adicionalmente, se comentarán las dificultades encontradas a lo largo de la realización del proyecto y por último, se propondrán futuras líneas de desarrollo que podrían utilizarse para dar continuidad al proyecto AAS11.

8.2 Conclusiones

Los objetivos planteados inicialmente, junto con el detalle de su cumplimiento aparecen comentados a continuación:

- Exposición de los conceptos de auditoría informática, control interno y su particularización sobre los entornos de bases de datos: este objetivo se considera cubierto en los Capítulos 2 (Auditoría Informática) y 3 (Auditoría de Bases de Datos) de este proyecto. El Capítulo 2 constituye una introducción donde se establece un contexto general en el que se lleva a cabo una descripción de los conceptos de auditoría y control interno. El Capítulo 3 tiene como finalidad realizar una descripción detallada del proceso de auditoría aplicable sobre bases de datos.

- Establecimiento de las bases de la arquitectura del sistema gestor de base de datos Oracle en su versión 11g: a la hora de realizar un proceso de auditoría se deben conocer las bases arquitectónicas y de funcionamiento de la aplicación sometida a análisis (en este caso Oracle 11g). Este ha sido el objetivo del Capítulo 4 (El Sistema Gestor de Bases de Datos Oracle 11g) que constituye un compendio de conceptos que describen de forma general las bases arquitectónicas de este sistema gestor y los principios de funcionamiento que lo dirigen.

- Aplicación de la metodología Scrum en el desarrollo de la aplicación AAS11: el paso previo para aplicar una metodología es conocerla con suficiente nivel de detalle para intentar obtener el máximo provecho posible. El Capítulo 5 (La Metodología Ágil Scrum) establece las motivaciones, fundamentos y bases de esta metodología con la finalidad de comprender su aplicación en el desarrollo de la aplicación AAS11. En el Capítulo 6 (Desarrollo de la aplicación AAS11) se expone de forma detallada el proceso de desarrollo utilizado, combinando la aplicación de la metodología junto con la utilización de herramientas de análisis y diseño habituales en el paradigma de la orientación a objetos.

Los subobjetivos asociados al desarrollo de la aplicación AAS11 junto con una descripción de su cumplimiento aparecen de forma detallada en el siguiente listado:

- Asistencia en la labor de auditoría sobre sistemas de información basados en el sistema gestor de bases de datos Oracle 11g: el objetivo principal del desarrollo ha sido la obtención de una herramienta que facilite la labor del auditor sobre entornos en los que se encuentre instalado el sistema gestor de base de datos Oracle 11g. Una de las ventajas principales que aporta esta herramienta es la combinación de cuestiones de auditoría con cuestiones de tuning permitiendo la obtención de informes completos que permitirán determinar la situación del sistema gestor de base de datos sometido a estudio.

- Presentar una interfaz amigable que permita la utilización simultánea de la aplicación desarrollada a usuarios con diferentes grados de conocimiento sobre la materia: uno de los elementos que se ha mantenido presente durante el diseño de la aplicación AAS11 ha sido intentar obtener una herramienta de manejo sumamente sencillo que permita a los usuarios obtener de forma rápida una evaluación de un sistema gestor de base de datos.

- Obtención de un informe de monitorización relativo al estado del sistema gestor de base de datos: la salida principal de la aplicación está constituida por un informe en formato Word que combina cuestiones de auditoría y tuning. La potencia de la aplicación AAS11 radica en que la evaluación de las cuestiones componen el informe de auditoría final está basada en consultas reales realizadas sobre la propia base de datos.

- Permitir el despliegue de la aplicación en entornos distribuidos utilizando el modelo cliente-servidor: el desarrollo de una aplicación web basada en un servidor de aplicaciones como Glassfish permitirá realizar una instalación en un servidor central y ser utilizada en un entorno multiusuario imponiendo la única condición de la disponibilidad, en cada equipo en el que se utilice, del navegador Google Chrome.

Las aportaciones, a nivel personal, durante la realización del proyecto fin de carrera han sido múltiples y engloban aspectos diferenciados que incluyen la ampliación de conocimientos de auditoría, utilización de nuevas metodologías de desarrollo del software y evaluación de herramientas de desarrollo y tecnologías empleadas actualmente en la implementación de aplicaciones:

- Ampliación de conocimientos de auditoría y utilización de los mismos dentro del sistema gestor de bases de datos Oracle 11g: una de las aportaciones principales del proyecto fin de carrera ha sido suministrar conocimiento sobre estándares, normativa vigente y buenas prácticas asociadas a los procesos de auditoría sobre una base de datos. Centrados en el ámbito del sistema gestor de bases de datos Oracle 11g la lista de evaluación CIS Becnhmark, analizada durante la fase de determinación de requisitos, supone un instrumento muy valioso de aplicación en sistemas de información construidos sobre esta plataforma.

- Utilización de la metodología ágil Scrum en contraposición a metodologías del desarrollo del software tradicionales: el proyecto fin de carrera ha sido aprovechado para utilizar la metodología de desarrollo ágil Scrum para poder evaluar sus características. La principal ventaja que he encontrado en la utilización de esta metodología es su capacidad de adaptación al cambio, ya que propone un proceso evolutivo y una comunicación constante entre usuarios y diseñadores sin generar un volumen de documentación intermedia que entorpezca esta comunicación. Hasta el instante anterior al desarrollo de este proyecto, he utilizado metodologías denominadas cómo pesadas (fundamentalmente variantes de la metodología Métrica v3 dependientes de la organización) en las que cualquier cambio está asociado a una serie de procesos formales y burocracia que suponen demoras de tiempo y pérdidas de eficiencia.

- Diseño y montaje de una arquitectura compleja utilizando herramientas de desarrollo y tecnologías actuales: el desarrollo de la aplicación AAS11 ha constituido una oportunidad para el estudio de nuevas tecnologías y para la implantación de una solución en la que el framework de Spring constituye su pilar principal.

8.3 Dificultades encontradas

Una de las principales dificultades encontradas en el desarrollo del proyecto ha sido la utilización de tecnologías como Spring, cuya curva de aprendizaje es elevada y cuyo uso implica el conocimiento de determinados conceptos como la programación orientada a aspectos (AOP) y la inyección de dependencias (DI). Adicionalmente, constituye un elemento complejo el manejo de la configuración y la asociación de objetos utilizando ficheros XML.

Con respecto a las herramientas utilizadas durante el desarrollo se podría ser particularmente crítico con el IDE (Spring tool Suite). Este IDE funciona muy lento en el intercambio de focos activos entre ventanas y en las labores de compilación y generación de ficheros WAR. Adicionalmente, la depuración ha sido tediosa y en ocasiones ha requerido del apoyo de un IDE algo más estandarizado como Eclipse en su versión Juno Service Release 2 para completar dicho proceso.

8.4 Futuras líneas de desarrollo

La especificación de requisitos que ha guiado la construcción de la aplicación AAS11 está estrechamente asociada a la lista de evaluación CIS Becnhmark y presenta las limitaciones de esta lista. Una primera aproximación a la extensión de esta herramienta podría incluir la implementación de los siguientes puntos:

- Definición e implementación de un sistema de evaluación de cada una de las cuestiones que permita obtener una valoración final (nota) sobre cada uno de los test realizados: podría establecerse un sistema de evaluación de cada una de las cuetiones de auditoría basándose en la asignación de una serie de pesos a cada cuestión. Estos pesos deberían definir la importancia de cada pregunta y poder ser utilizados para obtener un resultado final sobre cada test realizado, que posibilitará evaluar de forma inmediata si un sistema cumple con unos mínimos establecidos.

- Permitir la inclusión de test aplicables sobre otros sgbd: una siguiente versión de la herramienta podría permitir la realización de test de auditoría y tuning sobre otras plataformas como: SQL Server, DB2, Informix, Sybase y MySQL.

- Inclusión de módulos que reproduzcan intentos típicos de intrusión sobre sobre este tipo de sistemas para evaluar su nivel de seguridad: módulos que reproduzcan ataques de fuerza bruta sobre contraseñas, módulos para obtener acceso a información crítica utilizada para realizar un posterior ataque…

Glosario

AAPP	*Administraciones Públicas*
AAS11	*Automatic Audit System Oracle 11g*
ACID	*Atomicity Consistency Isolation Durability*
ADDM	*Automatic Database Diagnostic Monitor*
AMM	*Automatic Memory Management*
API	*Application Programming Interface*
AWR	*Automatic Workload Repository*
B2B	*Business to Business*
B2C	*Business to Consumer*
BDOO	*Base de Datos Orientada a Objetos*
BLOB	*Binary Large OBject*
CGI	*Common Gateway Interface*
COBIT	*Control OBjectives for Information and related Technology*
COSO	*Committee Of Sponsoring Organizations of the treadway commission*
CPU	*Central Processing Unit*
CSS	*Cascading Style Sheets*
DAO	*Data Access Object*
DBA	*DataBase Administrator*
DBMS	*DataBase Management System*
DDL	*Data Definition Language*
DES	*Data Encryption Standard*
DI	*Dependency Injection*
DISA	*Defense Information Systems Agency*
DML	*Data Manipulation Language*
DoD	*Department of Defense*
DSS	*Decision Support System*
EJB	*Enterprise Java Beans*
IFAC	*International Federation of Accountants*
ISACA	*Information Systems Audit and Control Association*
ITGI	*Information Technology Governance Institute*
JDBC	*Java DataBase Connectivity*
JDK	*Java Development Kit*
JSP	*Java Server Pages*
JSTL	*Java server pages Standard Tag Library*

LOPD	*Ley Orgánica de Protección de Datos*
LORTAD	*Ley ORgánica para el Tratamiento Automatizado de Datos*
LSSI	*Ley de Servicios de la Sociedad de la Información y comercio electrónico*
MVC	*Modelo Vista Controlador*
NOSQL	*Not Only Structured Query Languaje*
OAS	*Oracle Advanced Security*
OCDE	*Organización para la Cooperación y el Desarrollo Económico*
OEM	*Oracle Enterprise Manager*
OOXML	*Office Open XML*
PC	*Personal Computer*
PGA	*Program Global Area*
PL/SQL	*Programming Language / Structured Query Language*
POM	*Project Object Model*
RAC	*Real Application Cluster*
RMAN	*Recovery MANager*
RMI	*Remote Method Invocation*
ROI	*Return On Investment*
RSA	*Rivest Shamir Adleman*
SGA	*System Global Area*
SGBD	*Sistema Gestor de Base de Datos*
SGBDOO	*Sistema Gestor de Base de Datos Orientado a Objetos*
SGBDR	*Sistema Gestor de Base de Datos Relacional*
SHA	*Secure Hash Algorithm*
SQL	*Structured Query Language*
STIG	*Security Technical Implementation Guides*
TI	*Tecnología de la Información*
TO	*Transfer Object*
URL	*Uniform Resource Locator*
WAR	*Web Archive*
W3C	*World Wide Web Consortium*
XML	*eXtensible Markup Languaje*

Referencias

[Ach94] Acha Iturmendi JJ. *Auditoría Informática en la empresa*. Paraninfo 1994. Pp 35-37.

[Alo89] Alonso Rivas, Gonzalo. *Auditoría Informática*. Ediciones Diaz de Santos 1989. ISBN: 84-87189-13-X. Pp 97.

[AMP] *Apache Maven Project. Introduction to the POM.* Disponible en: http://maven.apache.org/guides/introduction/introduction-to-the-pom.html.Accedido en Septiembre 2012.

[ANSI73] ANSI, Quality Assurance Terms and Definitions,N45.2.10.1973,1973.

[Ben09] Ben-Natan,Ron.*How to Secure and Audit Oracle 10g and 11g*. Taylor & Francis Group,LLC 2009.ISBN: 978-1-4200-8412-2. Pp 59,99,188,223,255,315-335.

[Bel09] Bellas Permuy, Fernando. *Introducción al Diseño con Patrones*. Departamento de Tecnologías de la Información y las Comunicaciones. Universidad A. Coruña. Disponible en: http://www.tic.udc.es/~fbellas/teaching/pfc3/IntroPatrones.pdf. Accedido en Septiembre 2012.

[BMCS96] Bernal Montañés, Rafael y Coltell Simon, Óscar. *Auditoría de los sistemas de información.* Servicio de publicaciones de la Universidad Politécnica de Valencia 1996. ISBN: 84-7721-393-3. Pp. 136-145.

[BRJ99] Booch, Grady, Rumbaugh James, Jacobson Ivar. *El Lenguaje Unificado de Modelado*. Addison Wesley Iberoamericana 1999.ISBN:84-7829-028-01. Pp 191-199, 211-215.

[Bro75] Brooks, Frederick P. *The Mythical Man Month*. Addison Wesley 1975. ISBN: 0-201-83595-9.

[Burleson] Burleson Consulting. *Oracle Automatic Shared Memory Management.* Burleson Consulting (The Oracle of database Support). Disponible en: http://www.dba-oracle.com/t_automatic_shared_memory_management.htm. Accedido en Junio 2012.

[Cat02] Catálogo estándar ISO.*ISO/IEC 9796-2:2002 Information technology—Security techniques –Digital signature schemes giving message recovery– Part2:Integer factorization based mechanisms.* Abstract. Disponible en: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=35455. Accedido en Junio 2012.

[Cat08] Catálogo estándar ISO.*ISO/IEC 14888-1:2008 Information technology—Security techniques –Digital signatures with appendix– Part1:General.* Abstract. Disponible en: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=44226. Accedido en Junio 2012.

[Cat10] Catálogo estándar ISO.*ISO/IEC 11770-1:2010 Information technology—Security techniques – key management – Part1:Framekork*. Abstract. Disponible en: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=53456. Accedido en Junio 2012.

[Cec11] Ceccheti, Adam y otros autores.*Security Configuration Benchmark For Oracle Database Server 11g . Versión 1.1.0.* Leviathan Security Group, Inc. Diciembre 2011.The Center For Internet Security.Disponible en: http://www.cisecurity.org/bench_oracle.html. Accedido en Junio 2012.

[Cha08] Chan, Immanuel y otros autores. *Oracle Database Performance Tuning Guide 11g Release 1(11.1)*. Oracle. Julio 2008. ID: B28274-02. Disponible en: http://docs.oracle.com/cd/B28359_01/server.111/b28274.pdf. Pp 5-8 y 5-9. Accedido en Julio 2012.

[DA88] Delobel, C. y Adiba, M. *Bases de Datos y Sistemas Relacionales: Bases de Donnes*. Ediciones Omega, S.A.ISBN: 978-8428207584. Pp 47.

[Dee77] Deen S.M. *Fundamentals of Database Systems*. Hayden Book Company. Inc. Pp 32.

[Del98] Delgado Rojas, Xiomar. *Auditoría Informática*. Editorial Universidad Estatal a Distancia 1998. ISBN: 9977-64-937-5 667.463. Pp 89.

[DH90] DeGrace, Peter y Hulet Stahl, Leslie.*Wicked Problems, Righteus Solutions: A Catalog of Moderm Engineering Paradigms*. Prentice Hall, 1990.ISBN: 978-0135901267. Pp 115-127.

[EN08] Elsmari R y Navate S.B. *Fundamentals of Database Systems*. Pearson Addison-Wesley.ISBN: 978-8478290857. Pp 28.

[FL05] Flory André, Laforest Fredérique. *Les Bases de données Relationnelles*. Económica 2005.ISBN: 978-2717848717. Pp 38.

[Fra88] Frank, A. *Requirements for Database Management Systems for Large Spatial Databases.* Geol. Jb. Pp 58.

[Fro02] Froufe Quintas, Agustín. *JavaServer Pages. Manual de usuario y tutorial*. RA-MA 2002. ISBN: 84-7897-490-36. Pp 1-6.

[GL12] Gilbert, Set y Lynch Nancy A. *Perspectives on the CAP Theorem*. Disponible en: http://groups.csail.mit.edu/tds/papers/Gilbert/Brewer2.pdf. Accedido en Agosto 2012.

[GM01] Gutiérrez Rodriguez, Abraham y Martínez González, Raul. *XML a través de ejemplos*.RA-MA 2001.ISBN: 84-7897-455-5. Pp 165-167.

[Har10] Harrison, Guy.*Oracle Performance Survival Guide. A Systematic Approach to Database Optimization*. Prentice Hall 2010. ISBN: 978-0-13-701195-7. Pp 13-25.

[Her99] Herrera Joancomartí, Jordi.*Criptografía de clave pública*. Boletín del criptonomicón, Año I, número 44. *Disponible en:* http://www.iec.csic.es/criptonomicon/articulos/expertos44.html.Accedido en Junio 2012.

[Her11] Hernández Beatriz.*Export Oracle 10g/11g*. Disponible en: http://www.orasite.com/backup-de-base-de-datos/export-oracle-10g/11g. Accedido en Julio 2012.

[Heu09] Heurtel, Oliver.*Oracle 11g Administración*. Ediciones ENI 2009. ISBN: 978-2-7460-5169-0. Pp 15-44,443.

[HIB] Jboss Community. *Hibernate Validator*. Disponible en: http://www.hibernate.org/subprojects/validator.html.Accedido en Octubre 2012.

[Hum04] Humphreys, Ted. *ISO/IEC JTC1/SC27 Report-Globalization of Information Security*. Noviembre 2004. Disponible en: http://www.itsc.org.sg/pdf/1stmtg/Article9%20to%2011.pdf. Accedido en Junio 2012.

[IEEE08] IEEE.The IEEE P1363 Home Page. *Standard Specifications For Public-Key Cryptography*. Octubre 2008. Disponible en: http://grouper.ieee.org/groups/1363/. Accedido en Junio 2012.

[ISO/IEC05] ISO/IEC 27002. *Information Technology - Security techniques- Code of practique for information security management.*ISO (The International Organitation for Standardization) and IEC (the International Electrotechnical Commision).2005. Pp 39-114.

[ISO/IEC06] ISO/IEC 18028. *Information Technology - Security techniques- IT network security*.ISO (The International Organitation for Standardization) and IEC (the International Electrotechnical Commision). 2006.

[ITGI07] COBIT 4.1. *Framework, Control Objetives, Managements Guidelines, Maturity Models*. IT Governance Institute. EEUU 2007. Pp 33.

[Dan13] D'andrea, Mattew. *JfreeChart Tutorial*. Disponible en: http://project.management6.com/JFreeChart-Tutorial-download-w5448.pdf. Accedido en Febrero 2013.

[Lar02] Larman, Graig. UML y Patrones 2ª Edición.Pearson Educacion 2002. ISBN: 9788420534381.

[Lop12] López de Ipiña, Diego. *Bases de Datos No Relacionales (NOSQL).* Instituto de Tecnología de la Universidad de Deusto. Julio 2012. Disponible en: http://www.slideshare.net/dipina/nosql-cassandra-couchdb-mongodb-y-neo4j. Accedido en Agosto 2012.

[Mar75] Martin, James. *Computer Data-Base Organizations*. Prentice Hall 1975. Pp 17.

[MIT12] MIT (Massachusetts Institute of Technology).*Kerberos: The Network Authentication Protocol*. Febrero de 2012. Disponible en: http://web.mit.edu/kerberos/. Accedido en Junio 2012.

[MP99] de Miguel Castaño, Adoración y Piattini Velthuis, Mario Gerardo. *Fundamentos y modelos de bases de datos, 2 Edición*.RA-MA 1999.ISBN: 84-7897-361-3. Pp 25-28.

[MPM99] de Miguel Castaño, Adoración, Piattini Velthuis, Mario y Marcos Martinez, Esperanza. *Diseño de Bases de Datos relacionales*.RA-MA 1999.ISBN: 84-7897-385-0. Pp 23-41.

[OCDE97] OCDE (Organización para la Cooperación y el Desarrollo Económico). *Directrices para una política criptográfica. Recomendación del consejo en relación con las directrices para una política criptográfica*. Marzo de 1997. Disponible en: http://administracionelectronica.gob.es/recursos/pae_000005909.pdf. Accedido en Junio 2012.

[ONU77] ONU – *Conférence des statisticiens européens sur le traitement électronique de l'information*.Ginebra, 21-25 Febrero. 1977.

[Ora07] Oracle. *Informe Técnico sobre Oracle Advanced Security*. Informe ejecutivo de Oracle. Junio de 2007. Disponible en: http://www.oracle.com/technetwork/es/database/enterprise-edition/documentation/oracle-advanced-security-11g-426368-esa.pdf. Accedido en Junio 2012. Pp 11.

[Orabase1] Oracle-Base.*Automatic Database Diagnostic Monitor (ADDM) in Oracle Database 10g.*Disponible en: http://www.oracle-base.com/articles/10g/automatic-database-diagnostic-monitor-10g.php. Accedido en Julio 2012.

[Orabase2] Oracle-Base. *AWR Baseline Enhancements in Oracle Database 11g Release 1*.Disponible en: http://www.oracle-base.com/articles/11g/awr-baseline-enhancements-11gr1.php. Accedido en Julio 2012.

[OraDocs] Oracle Documents. *BLOB Data Type.* Disponible en: http://docs.oracle.com/javadb/10.8.1.2/ref/rrefblob.html. Accedido en Enero 2013.

[Orasite10] *Administración de usuarios en Oracle.*Diciembre de 2010 Disponible en: http://www.orasite.com/administracion-de-oracle/administracion-de-usuarios-en-oracle. Accedido en Junio 2012.

[Pav08] Pavón Mestras, Juan. *El patrón MVC*. Departamento de Ingenieria del Software e Inteligencia Artificial. Universidad Complutense de Madrid 2008. Disponible en: http://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf. Accedido en Noviembre 2012.

[PMM+05] Pons Capote, Olga, Marín Ruíz, Nicolas, Medina Rodriguez, Juan Miguel y otros autores. *Introducción a las Bases de Datos*. Paraninfo 2005. ISBN: 84-9732-396-3. Pp 15-16.

[POI13] The Apache POI project. *The Java API for Microsoft Documents*. Disponible en: http://poi.apache.org/. Accedido en Marzo 2013.

[PPP+08] Piattini Velthuis, Mario; del Peso Navarro, Emilio; del Peso Ruiz, Mar y otros autores. *Auditoría de Tecnologías y Sistemas de Información*. RA-MA 2008. ISBN: 978-84-7897-849-6. Pp 5-25.

[RW73] Rittel, Horst y Webber, Melvin.*Dilemmas in a General Theory of Planning. Policy Sciences*. Elsevier Scientific Publishing Company,1973. Pp 155-169.

[RWR+00] Rigney, C;Willens Livingston, S; Rubens Merit, A y Simpson,W. *Request for comments 2865. Remote Authentication Dial In User Service (RADIUS)*. The Internet Society. Junio 2000. Disponible en: http://www.ietf.org/rfc/rfc2865.txt. Accedido en Junio 2012.

[SB01] Schwaber, Ken y Beedle, Mike.*Agile Software Development with Scrum*. Prentice Hall,2001.ISBN: 978-0130676344. Pp 73-120.

[TB10] Thomas, Biju y Bryla, Bob. *Oracle 11g,. DBA Fundamentals 1*. Sybex 2010. ISBN: 0-7821-4063-7. Pp 4-14.

[tiles] *Apache Software Foundation. Apache Tiles.* Disponible en: http://tiles.apache.org/. Accedido en Septiembre 2012.

[TN86] Takeuchi, Irotaka y Nonaka, Ikujiro. *The New New Product Development Game*. Harvard Business Review, Enero 1986.

[Wal11] Walls, Craig. *Spring Tercera Edición*. Anaya Multimedia 2011. ISBN: 978-84-415-3041-6. Pp 31-45,100-110,218-225.

Anexo 1

En este anexo se incluye un ejemplo de informe de auditoría/tuning generado de forma automática por la aplicación AAS11 partiendo de los siguientes supuestos:

- Tipo de sistema operativo: UNIX.
- Versión de Oracle: 11.2.0.0.0.
- Productos de base instalados:
    - NLSRTL: versión 11.2.0.1.0.
    - Oracle Database 11g Enterprise Edition: versión 11.2.0.1.0.
    - PL/SQL: versión 11.2.0.1.0.
    - TNS for 64-bit: versión 11.2.0.1.0.

El documento generado consta de una serie de cuestiones de auditoría y cuestiones de tuning organizadas en las siguientes secciones y apartados:

- Secciones:
    - Configuración específica de sistema operativo.
    - Instalación y parches.
    - Directorio Oracle y permisos sobre ficheros.
    - Parámetros de configuración de Oracle.
    - Configuración específica sobre cifrado.
    - Arranque y parada.
    - Backup y recuperación ante desastres.
    - Parametrización de puesta a punto de perfiles.
    - Parametrización de acceso al perfil de usuario.
    - Enterprise Manager/Grid Control/Agentes.
    - Elementos relevantes para subsistemas específicos.
    - Políticas generales y procedimientos.
    - Políticas de auditoría y procedimientos.
    - Apéndice A. Configuración adicional.

- Apartados:

    - Espacio de almacenamiento: Tablespaces, segmentos.
    - Distribución del almacenamiento en memoria.
    - Objetos de la base de datos.

Aquellas cuestiones de auditoría que tienen asociada una consulta SQL son especialmente útiles, puesto que esta consulta determinará la respuesta que debe asignarse a la cuestión. En estas cuestiones se adjuntará la tabla resultado obtenida tras la realización de la consulta.

Las cuestiones de tuning permitirán determinar el estado de algunos elementos que componen la base de datos como: tablespaces, segmentos, memoria, tablas, índices… Estas cuestiones tienen asociada una consulta SQL que suministra dos columnas. La primera columna contiene aquellos elementos sobre los que se quiere extraer la información. La segunda columna contiene el valor asociado a cada elemento. Este valor será el que se utilice para llevar a cabo la representación en el gráfico seleccionado (diagrama de barras o diagrama de tarta). En estas cuestiones se adjuntará, tanto el resultado de la tabla obtenida tras la realización de la consulta como el gráfico generado con estos datos.

El documento que aparece a continuación, se ha mantenido con el formato original generado por la aplicación AAS11.

INFORME DE AUDITORIA:

Informe 27

Fecha: 27/05/2013

1 Configuración específica de sistema operativo

13.- ¿Está habilitado el bloqueo de cuentas de usuario?

Comentario: el bloqueo de cuentas de usuario podrá impedir a los atacantes utilizar estas cuentas para ataques de fuerza bruta.

Está habilitado el bloqueo de cuentas de usuario.

14.- ¿Se han verificado los permisos sobre la totalidad de ficheros de aplicación?

Comentario: permitir accesos a ficheros binarios que interactúan con la base de datos añade riesgos innecesarios e incrementa la superficie de ataque sobre la base de datos.

Se han revisado los permisos asociados con la totalidad de ficheros de aplicación.

2 Instalación y parches

15.- ¿Se ha asegurado de que otros usuarios se han conectado mientras se instalaba Oracle 11g?

Comentario: la instalación de Oracle puede crear ficheros, cuentas, o parametrizaciones con privilegios. Un atacante podría aprovecharlo para comprometer la integridad del sistema.

No. Ubique las variables de entorno $SMTP y $TMPDIR en un directorio con acceso limitado al propietario del software de Oracle y al grupo ORA_INSTALL. Instale el software de Oracle sin conexión a red.

16.- ¿Se ha asegurado de que se está utilizando la última versión de software y han sido aplicados los últimos parches de Oracle Metalink?

Comentario: la utilización de versiones de software desfasadas o no actualizadas sitúa a la base de datos Oracle y al sistema anfitrión en una posición de riesgo.

No. Compruebe las direcciones de Oracle para versiones de software: http://www.Oracle.com/technology/software/index.html. Para parches: http://metalink.oracle.com/metalink/plsql/ml2_gui.startup.

17.- ¿Se ha comprobado que únicamente se han utilizado los componentes de Oracle necesarios para el entorno de instalación?

Comentario: instalar componentes que no serán utilizados incrementa la superficie de ataque del servidor de bases de datos.

No. Instale únicamente los componentes necesarios para satisfacer los requisitos de operación. Elimine componentes instalados durante una instalación previa y que no son necesarios.

18.- ¿Se ha eliminado del sistema la utilidad tkprof?

Comentario: esta utilidad debe ser eliminada de los entornos de producción. Es una potente herramienta a disposición del atacante para permitir la utilización de bases de datos en ejecución.

No. Si no es posible eliminar esta utilidad establezca permisos 0750 o más limitados en sistemas UNIX. En sistemas Windows, limite el acceso a aquellos usuarios autorizados.

19.- ¿Se ha cambiado el nombre por defecto del listener?

Comentario: el listener no debe denominarse con el nombre por defecto. Debe seleccionar un nombre distinto para impedir su localización.

No. Para realizar el cambio de nombre edite: $ORACLE_HOME/network/admin/listener.ora

20.- ¿Se han utilizado en el listener direcciones IP en lugar de nombres de Host?

Comentario: en el listener deben utilizarse direcciones IP en lugar de nombres de host. Esto evita problemas de ataques asociados con el DNS.

No. Para cambiar esto edite: $ORACLE_HOME/network/admin/listener.ora. Reemplace en este archivo nombres de host por direcciones IP.

21.- ¿Se ha deshabilitado Otrace?

Comentario: Otrace puede exponer información sensible utilizable por un atacante.

No. Elimine los ficheros .dat asociados a Otrace situados en $ORACLE_HOME/otrace/admin. Este directorio está asociado a la in stalación de Enterprise Manager Grid Controller.

22.- ¿Está utilizando autenticación a través de SSOO en el password del listener?

Comentario: es más seguro utilizar autenticación del sistema operativo para establecer el password del listener. Esto permitirá habilitar la administración remota del listener.

Se ha utilizado autenticación de Sistema Operativo en el establecimiento del password del listener.

24.- Realice las siguientes acciones en orden de prioridad sobre las cuentas por defecto: 1- Eliminar el usuario. 2- Bloquear la cuenta de usuario. 3- Cambiar la password por defecto.

Comentario: la instalación de Oracle por defecto bloquea y expira las cuentas instaladas. Estas cuentas deben permanecer bloqueadas y expiradas a menos que sea estrictamente necesario.

USERNAME
DIP
XS$NULL
MDSYS
SPATIAL_WFS_ADMIN_USR
OUTLN
OLAPSYS
SPATIAL_CSW_ADMIN_USR
OWBSYS
ORACLE_OCM
EXFSYS
SCOTT
ORDSYS
ORDPLUGINS
MDDATA
PM
APPQOSSYS
XDB
ORDDATA
IX
BI
WMSYS
SI_INFORMTN_SCHEMA

Comentario consulta: un resultado de esta SELECT identifica cada cuenta que tiene un password por defecto.

El sistema es susceptible de recibir algún ataque a partir de las cuentas de por defecto. Debe llevar a cabo las acciones que se recomiendan y bloquear o caducar dichas cuentas.

25.- ¿Ha eliminado Oracle Enterprise Manager en el caso de que no vaya a ser utilizado?

Comentario: eliminar Oracle Enterprise Manager reduce la superficie de ataque del sgbdr.

Se ha procedido a la eliminación de Oracle Enterprise Manager por lo que se ha reducido la superficie de ataque.

26.- ¿Ha cambiado los puertos por defecto de acceso al listener?

Comentario: los puertos por defecto de acceso al listener son utilizados en ataques automáticos para verificar las aplicaciones que se están ejecutando en un servidor.

No se han cambiado los puertos por defecto de acceso al listener. Para modificarlos edite el fichero listener.ora y cambie la configuración de puertos.

27.- ¿Ha substituido las password de las cuentas por defecto instaladas por aplicaciones de terceros por password fuertes?

Comentario: cuando instala ciertas aplicaciones de terceros, estas aplicaciones generan cuentas por defecto conocidas en la base de datos.

USERNAME
DIP
XS$NULL
MDSYS
SPATIAL_WFS_ADMIN_USR
OUTLN
OLAPSYS
SPATIAL_CSW_ADMIN_USR
OWBSYS
ORACLE_OCM
EXFSYS
SCOTT
ORDSYS
ORDPLUGINS
MDDATA
PM
APPQOSSYS
XDB
ORDDATA
IX
BI
WMSYS
SI_INFORMTN_SCHEMA

Comentario consulta: un resultado de esta SELECT identifica cada cuenta que tiene un password por defecto

No. Debe cambiar la password de estas cuentas o llevar a cabo un bloqueo de las mismas para impedir su utilización.

28.- ¿Ha utilizado un nombre distinto al nombre del servicio o al nombre de la instancia que suministra Oracle por defecto?

Comentario: no debe utilizar el nombre por defecto de la instancia o del servicio suministrado en la instalación de Oracle. Este nombre (ORCL) podría ser utilizado en un posible ataque automático.

Sí. El nombre no coincide con el nombre de instancia por defecto y por tanto será más complicado identificarla.

29.- ¿Ha asignado al propietario de la cuenta del software de Oracle un nombre distinto de "Oracle"?

Comentario: no debe denominar al propietario de la cuenta de software de Oracle con el nombre de "Oracle" puesto que este dato podría ser utilizado en un ataque de fuerza bruta.

Sí. Utilizar un nombre distinto al que se presupone que tendrá la cuenta del propietario del software dificultará las posibilidades de acceso a la misma en un ataque de estas características.

3 Directorio de Oracle y permisos sobre ficheros

31.- ¿Ha configurado y verificado el propietario del directorio $ORACLE_HOME/bin?

Comentario: la totalidad de los ficheros de este directorio deben pertenecer a la cuenta de software de Oracle para evitar comprometer el resto del sistema en el caso de que esta cuenta se comprometa.

Sí. La totalidad de ficheros binarios alojados en el directorio %ORACLE_HOME/bin pertenecen al usuario asociado al software de Oracle.

32.- ¿Ha establecido permisos del tipo 0755 o inferiores sobre los ficheros del directorio $ORACLE_HOME/bin?

Comentario: permisos incorrectos sobre los ficheros de este directorio podrían permitir a un atacante remplazar un fichero binario por una versión maliciosa.

Sí. Con esta mascara de permisos evitará a determinados usuarios la posibilidad de substituir determinados ficheros.

33.- ¿Ha establecido permisos del tipo 0750 o inferiores sobre los ficheros del directorio $ORACLE_HOME exceptuando $ORACLE_HOME/bin?

Comentario: permisos incorrectos sobre los ficheros de este directorio podrían permitir a un atacante ejecutar o remplazar un fichero binario por una versión maliciosa.

No. Para cambiar estos permisos ejecute: chmod 750 $ORACLE_HOME/bin/*.

34.- ¿Ha establecido el valor de umask en el fichero .profile al valor 022?

Comentario: se debe establecer el valor de umask a 022 antes de instalar Oracle. Un valor incorrecto podría permitir establecer permisos incorrectos y problemas en el sistema de ficheros del servidor de Oracle.

No. Debe establecer el valor de umask en el fichero .profile a 022 para el propietario del software de Oracle antes de la instalación de Oracle.

35.- ¿Ha restringido y verificado permisos sobre el fichero init.ora?

Comentario: los permisos sobre los ficheros deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Conceder los permisos correctos sobre el grupo y el usuario debe realizar estas acciones: chgroup oracle_grp init.ora, chown oracleuser init.ora, chmod 644 init.ora.

36.- ¿Ha restringido y verificado permisos sobre el fichero spfile.ora?

Comentario: los permisos sobre los ficheros deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Conceder los permisos correctos sobre el grupo y el usuario debe realizar estas acciones: chgroup oracle_grp spfile.ora, chown oracleuser spfile.ora, chmod 640 spfile.ora.

37.- ¿Ha restringido y verificado permisos sobre los ficheros datafiles de la base de datos?

Comentario: los permisos deben estar limitados al propietario del software de Oracle y al grupo dba. Un usuario sin privilegios podría alterar los datafiles y comprometer la seguridad del servidor Oracle.

No. Para cambiar los permisos en sistemas en sistemas UNIX: chown oracleuser $ORACLE_HOME/dbs/* chgrp oraclegroup $ORACLE_HOME/dbs/*.

38.- ¿Ha restringido y verificado permisos del fichero referenciado por el parámetro ifile en init.ora?

Comentario: los permisos deben estar limitados al propietario del software de Oracle y al grupo dba. Si se utiliza la funcionalidad ifile, el fichero referenciado por ifile debe contemplar estas limitaciones.

Sí. Se han verificado los permisos sobre el fichero referenciado por el parámetro ifile en el caso de que se utilice esta funcionalidad.

39.- ¿Ha comprobado la configuración del parámetro audit_file_dest en el fichero init.ora?

Comentario: el destino del fichero de auditoría debe ser establecido sobre un directorio válido propiedad del usuario de Oracle y con únicamente permisos de lectura/escritura para este usuario.

Sí. Se ha verificado tanto el propietario del fichero como los permisos asignados y son correctos.

40.- ¿Ha comprobado la configuración del parámetro diagnostic_dest en el fichero init.ora?

Comentario: el destino para el fichero asociado al user dump debe ser un fichero con permisos restringidos al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX:chmod 660 diag_file, chown oracleuser.oraclegroup diag_file.

41.- ¿Ha comprobado la configuración del parámetro control_files en el fichero init.ora?

Comentario: estos permisos deben ser limitados al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX:chmod 640 control_file, chown oracleuser.oraclegroup control_file.

42.- ¿Ha comprobado la configuración del parámetro log_archive_dest_n en el fichero init.ora?

Comentario: los permisos estarán limitados al propietario del software Oracle y al grupo dba. Para configuraciones en las que varios grupos acceden al directorio, se utilizarán listas de control de acceso.

No. Para cambiar los permisos en sistemas UNIX: chmod 750 file_file dest, chown oracleuser.oraclegroup \log_file_dest.

43.- ¿Ha establecido y comprobado permisos sobre el directorio $ORACLE_HOME/network/admin?

Comentario: los permisos para todos los ficheros contenidos en este directorio deben estar limitados al propietario del software de Oracle y al grupo dba.

Sí. Se han verificado los permisos y el propietario de los ficheros contenidos en $ORACLE_HOME/network/admin y son correctos.

44.- ¿Ha verificado y establecido permisos de lectura para todo el mundo sobre el fichero sqlnet.ora?

Comentario: el fichero sqlnet.ora contiene los ficheros de configuración para la comunicación entre el usuario y el servidor incluyendo el nivel de encriptación requerido.

Sí. Se han verificado los permisos sobre el fichero sqlnet.ora comprobando que se han asignado permisos de lectura a todos los usuarios.

45.- ¿Ha comprobado el valor del parámetro log_directory_client en el fichero sqlnet.ora?

Comentario: el valor del parámetro log_directory_client debe contener un directorio propietario de cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y para el grupo dba.

Sí. Se ha comprobado el valor del parámetro y contiene un directorio con la asignación de permisos correcta.

46.- ¿Ha comprobado el valor del parámetro log_directory_server en el fichero sqlnet.ora?

Comentario: el valor del parámetro log_directory_server debe contener un directorio propietario de cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y el grupo dba.

Sí. Se ha comprobado el valor del parámetro y contiene un directorio con la asignación de permisos correcta.

47.- ¿Ha comprobado el valor del parámetro trace_directory_client en el fichero sqlnet.ora?

Comentario: el valor del parámetro trace_directory_client debe contener un directorio propietario de cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y el grupo dba.

Sí. Se ha comprobado el valor del parámetro y contiene un directorio con la asignación de permisos correcta.

48.- ¿Ha comprobado el valor del parámetro trace_directory_server en el fichero sqlnet.ora?

Comentario: el valor del parámetro trace_directory_server debe contener un directorio propietario de cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y el grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chmod 640 trace_directory_server, chown oracleuser.oraclegroup log_directory_server.

49.- ¿Ha establecido y verificado permisos sobre el fichero listener.ora?

Comentario: los permisos sobre este fichero deben estar restringidos al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chmod 640 $ORACLE_HOME/network/admin/listener.ora.

50.- ¿Ha comprobado el valor del parámetro log_file_listener en el fichero listener.ora?

Comentario: el valor del parámetro log_file_listener debe contener un directorio propietario de la cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software de Oracle y el grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chmod 640 $ORACLE_HOME/network/log/listener.log, chown oracleuser.oraclegroup $ORACLE_HOME/network/log/listener.log.

51.- ¿Ha comprobado el valor del parámetro trace_directory_listener_name en el fichero listener.ora?

Comentario: el valor del parámetro trace_directory_listener debe contener un directorio propietario de la cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y el grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chmod 660 trace_dir, chown oracleuser.oraclegroup trace_dir.

52.- ¿Ha comprobado el valor del parámetro trace_file_listener_name en el fichero listener.ora?

Comentario: el valor del parámetro trace_file_listener_name debe contener un directorio propietario de la cuenta Oracle y permisos sólo de lectura/escritura para el propietario del software Oracle y el grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chown oracleuser.oraclegroup $ORACLE_HOME/network/trace, chmod 660 $ORACLE_HOME/network/trace.

53.- ¿Ha establecido y verificado permisos sobre el fichero ejecutable asociado al sqlplus?

Comentario: los permisos asociados al fichero ejecutable del sqlplus deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chown oracleuser.oraclegroup sqlplus, chmod 750 sqlplus.

54.- ¿Ha establecido y verificado permisos sobre el fichero .htaccess?

Comentario: los permisos asociados al fichero .htaccess deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chown oracleuser.oraclegroup .htaccess, chmod 644 .htaccess.

55.- ¿Ha establecido y verificado permisos sobre el fichero dads.conf?

Comentario: los permisos asociados al fichero dads.conf deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chown oracleuser.oraclegroup dads.conf, chmod 644 dads.conf.

56.- ¿Ha establecido y verificado permisos sobre el fichero xsqlconfig.xml?

Comentario: los permisos asociados al fichero xsqlconfig.xml deben estar limitados al propietario del software de Oracle y al grupo dba.

No. Para cambiar los permisos en sistemas UNIX: chown oracleuser.oraclegroup xsqlconfig.xml, chmod 640 XSQLConfig.xml.

4 Parámetros de configuración de Oracle

57.- ¿Ha comprobado que el valor del parámetro _trace_files_public se ha establecido a FALSE en el fichero init.ora?

Comentario: este parámetro permite prevenir que los usuarios tengan la posibilidad de leer trazas que contengan información sensible sobre la ejecución de la instancia Oracle.

No. Establezca el valor del parámetro _trace_files_public a FALSE.

58.- ¿Ha comprobado que el valor del parámetro global_names se ha establecido a TRUE en el fichero init.ora?

Comentario: este parámetro permite garantizar que el sgbd comprobará el nombre de la conexión de base de datos remota (databse link) es el mismo que el de la base de datos remota.

No. Establezca el valor del parámetro global_names a TRUE.

59.- ¿Ha comprobado que el valor del parámetro remote_os_authent se ha establecido a FALSE en el fichero init.ora?

Comentario: esta característica está obsoleta, aunque se mantiene para garantizar la compatibilidad hacia atrás. Si está configuración se utiliza es recomendable que el valor de este parámetro esté a falso.

No. Establezca el valor del parámetro remote_os_authent a FALSE.

60.- ¿Ha comprobado que el valor del parámetro remote_os_roles se ha establecido a FALSE en el fichero init.ora?

Comentario: este parámetro reducirá el riesgo de que se utilicen técnicas de suplantación de identidad.

No. Establezca el valor del parámetro remote_os_roles a FALSE.

61.- ¿Ha comprobado que el valor del parámetro remote_listener se ha establecido como cadena vacía en el fichero init.ora?

Comentario: el valor de este parámetro previene del uso del listener en una máquina remota separada de la instancia de base de datos.

No. Establezca el valor del parámetro remote_os_listener a cadena vacía.

62.- ¿Ha comprobado que el valor del parámetro audit_trail se ha establecido con uno de los siguientes valores: OS, DB, DB_EXTENDED, XML o XML_EXTENDED en el fichero init.ora?

Comentario: este parámetro permite garantizar que se utilizarán las características básicas de auditoría. Se recomienda el valor OS ya que reduce la posibilidad de sufrir un ataque de denegación de servicio.

No. Establezca el valor del parámetro audit_trail en uno de los valores recomendados siendo OS el valor preferido.

63.- ¿Ha comprobado que el valor del parámetro os_authent_prefix se ha establecido como cadena vacía en el fichero init.ora?

Comentario: los roles de SO están sujetos a un control externo fuera del ámbito de la base de datos. Las responsabilidades de los DBAs y de los administradores de la base de datos deben estar separadas.

No. Establezca el valor del parámetro os_authent_prefix a cadena vacía.

64.- ¿Ha comprobado que el valor del parámetro os_roles se ha establecido a FALSE en el fichero init.ora?

Comentario: el parámetro os_roles permite crear grupos externos utilizados para gestionar los roles de la base de datos. Esto puede provocar permisos inconsistentes y heredados.

No. Establezca el valor del parámetro OS_ROLES a FALSE.

65.- ¿Ha evitado el uso de parámetros utl_file_dir en el fichero init.ora?

Comentario: no utilice el parámetro utl_file_dir ya que los directorios creados mediante este parámetro pueden ser leídos y escritos por todos los usuarios. Utilice CREATE DIRECTORY para llevar a cabo esta labor.

No. Evite el uso de parámetros utl_file_dir. Para especificar directorios utilice CREATE DIRECTORY.

66.- ¿Ha comprobado el valor TRUE del parámetro sql92_security en el fichero init.ora?

Comentario: obligar a cumplir el requerimiento de que los usuarios deben disponer del privilegio de SELECT sobre una tabla para ejecutar UPDATES y DELETES utilizando la clausula WHERE.

No. Debe establecer el valor de esta parámetro a TRUE realizando la acción "Set sql92_security=TRUE".

67.- ¿El valor del parámetro admin_restrictions_listener_name es igual a ON en listener.ora?

Comentario: este valor permitirá que el listener rechace "set commands" que podrían alterar el valor de sus parámetros sin obligar a reiniciarlo.

No. Establezca el valor de este parámetro a ON utilizando el comando "Set admin_restrictions_listener_name=ON".

68.- ¿El valor del parámetro loggin_listener está establecido a ON en el fichero listener.ora?

Comentario: registrar todos los intentos de acceso al listener a través de un registro de auditoría permitirá identificar posibles ataques sobre la configuración del listener.

No. Establezca el valor del parámetro loggin_listener a ON utilizando el comando "Set loggin_listener=ON".

69.- ¿El valor del parámetro tcp.validnode_checking es igual a YES en el fichero sqlnet.ora?

Comentario: este parámetro habilita al listener la posibilidad de comprobar conexiones entrantes coincidentes con la lista de nodos incluidos y excluidos

No. Establezca el valor de este parámetro a YES utilizando el comando "Set tcp.validnode_checking=YES sobre Sqlnet.ora.

70.- ¿Ha asignado al parámetro tcp.invited_nodes valores válidos?

Comentario: este parámetro permite crear una lista de nodos de confianza que podrán conectar con el listener. El valor del parámetro exluded_nodes será ignorado si se ha asignado valor a este parámetro.

No. Establezca el valor del parámetro tcp.invited_nodes cómo muestra el siguiente ejemplo tcp.invited_nodes={10.1.1.1, 10.1.1.2}.

71.- ¿Ha bloqueado el acceso a cuentas para los propietarios de esquemas de aplicación?

Comentario: se debe bloquear el acceso para propietarios de esquemas de aplicación. Los usuarios no deben conectarse a la base de datos con privilegios de propietario del esquema de aplicación.

USERNAME	ACCOUNT_STATUS
SYS	OPEN
SYSTEM	OPEN
DBSNMP	OPEN
SYSMAN	OPEN
AAS11	OPEN
MGMT_VIEW	LOCKED
OUTLN	EXPIRED & LOCKED
FLOWS_FILES	EXPIRED & LOCKED
MDSYS	EXPIRED & LOCKED
ORDSYS	EXPIRED & LOCKED
EXFSYS	EXPIRED & LOCKED
WMSYS	EXPIRED & LOCKED
APPQOSSYS	EXPIRED & LOCKED
APEX_030200	EXPIRED & LOCKED
OWBSYS_AUDIT	EXPIRED & LOCKED
ORDDATA	EXPIRED & LOCKED
CTXSYS	EXPIRED & LOCKED
ANONYMOUS	EXPIRED & LOCKED
XDB	EXPIRED & LOCKED
ORDPLUGINS	EXPIRED & LOCKED
OWBSYS	EXPIRED & LOCKED
SI_INFORMTN_SCHEMA	EXPIRED & LOCKED
OLAPSYS	EXPIRED & LOCKED
SCOTT	EXPIRED & LOCKED
ORACLE_OCM	EXPIRED & LOCKED
XS$NULL	EXPIRED & LOCKED
BI	EXPIRED & LOCKED
PM	EXPIRED & LOCKED
MDDATA	EXPIRED & LOCKED
IX	EXPIRED & LOCKED
SH	EXPIRED & LOCKED
DIP	EXPIRED & LOCKED
OE	EXPIRED & LOCKED

APEX_PUBLIC_USER	EXPIRED & LOCKED
HR	EXPIRED & LOCKED
SPATIAL_CSW_ADMIN_USR	EXPIRED & LOCKED
SPATIAL_WFS_ADMIN_USR	EXPIRED & LOCKED

Comentario consulta: en esta consulta debe anotar aquellas cuentas que aparecen abiertas "OPEN" y bloquearlas utilizando la siguiente sentencia: ALTER USER <USERNAME> ACCOUNT LOCK PASSWORD EXPIRE.

No. Para remediar esta situación ejecute: ALTER USER <USERNAME> ACCOUNT LOCK PASSWORD EXPIRE

72.- ¿Ha establecido el valor del parámetro SECURE_PROTOCOL_listener_name a (TCP, IPC)?

Comentario: debe asegurarse de que cada petición de administración se acepta solamente sobre un protocolo seguro. Si solamente se permite IPC o TCP limite la configuración de este parámetro a estos valores.

No. Para modificarlo edite el fichero listener.ora y cambie el valor del parámetro SECURE_PROTOCOL_listener_name a (TCP,IPC).

5 Configuración específica sobre cifrado

73.- ¿Ha revisado los requerimientos de integridad y confidencialidad?

Comentario: solo debe utilizarse OAS si no existe una política de integridad/cifrado. Por ejemplo utilizar IPSec u otros protocolos permite suministrar servicios de integridad y confidencialidad.

No. Debe revisar los requerimientos de integridad y confidencialidad para determinar si es necesaria la implantación de OAS.

74.- ¿Ha establecido el valor del parámetro SQLNET.ENCRYPTION_SERVER al valor REQUIRED en sqlnet.ora?

Comentario: el valor de este parámetro asegurará que independientemente de la configuración del usuario, si la comunicación se lleva a cabo debe estar cifrada.

No. Para establecer el valor de este parámetro debe editar el fichero sqlnet.ora, localizar el parámetro SQLNET.ENCRYPTION_SERVER y asignarle el valor REQUIRED.

75.- ¿Ha establecido el valor del parámetro SQLNET.ENCRYPTION_SERVER a valor REQUIRED? en el fichero sqlnet.ora?

Comentario: la comunicación es solamente posible sobre la base de un acuerdo entre el cliente y el servidor con respecto al cifrado utilizado. Para asegurar una comunicación cifrada, debe usar el valor REQUIRED.

No. Para remediar esta situación debe editar el fichero sqlnet.ora y modificar el valor del parámetro SQLNET.ENCRYPTION_CLIENT al valor REUIRED.

76.- ¿Ha establecido el valor del parámetro SSLFIPS_140 a valor TRUE en el fichero fips.ora?

Comentario: con la finalidad de cumplir el estándar FIPS 140-2 el valor de este parámetro debe estar establecido a TRUE. El valor por defecto está establecido a FALSE.

No. Para modificar el valor de este parámetro edite el fichero fips.ora y cambie el valor del parámetro SSLFIPS_140 a valor TRUE. Este valor no es modificable utilizando el asistente Oracle Net.

77.- ¿Ha establecido el valor del parámetro SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER al valor SHA1 en el fichero sqlnet.ora?

Comentario: si es posible, utilice SHA1 en vez de MD5. En MD5 se han identificado debilidades y es posible que se produzcan colisiones. El uso de SHA-1 está recomendado.

No. Para establecer el valor de este parámetro edite el fichero sqlnet.ora, localice el parámetro SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER y asigne el valor SHA1.

78.- ¿Ha establecido un método de configuración para Oracle Wallet?

Comentario: asegúrese de que únicamente el usuario adecuado de Oracle tenga acceso a Oracle Wallet. La cuenta seleccionada debe tener acceso a Oracle Wallet.

No. Debe seleccionar una cuenta única de usuario para disponer de acceso a Oracle Wallet.

79.- ¿Ha eliminado los Certificados de Autoridad (CAs) que no sean necesarios?

Comentario: utilice únicamente aquellos Certificados de Autoridad necesarios para clientes y servidores.

No. Debe asegurarse de que únicamente está utilizando los Certificados de Autoridad necesarios.

80.- ¿Se ha asegurado de comprobar la autenticidad de las firmas cuando añade Certificados de Autoridad?

Comentario: cuando añade Certificados de Autoridad debe comprobar la firma para verificar el certificado. Fallos en la comprobación de firma en los Certificados de Autoridad puede conducir a fallos de seguridad.

No. Debe asegurarse de comprobar la autenticidad de las firmas cuando añade Certificados de Autoridad.

81.- ¿Se ha asegurado de establecer el valor del parámetro SSL_VERSION en el fichero sqlnet.ora al valor 3.0?

Comentario: debe utilizar la versión más actualizada del protocolo SSL. Sobre versiones anteriores se han notificado errores y fallos de seguridad. En ningún caso establezca el valor de este parámetro a "Any".

No. Debe asegurarse de asignar al parámetro SSL_VERSION el valor de la última versión del protocolo SSL. En ningún caso el valor de este parámetro debe ser "Any".

82.- ¿Ha eliminado el privilegio de ejecución PUBLIC de la tabla DBMS_OBFUSCATION_TOOLKIT?

Comentario: el DBMS_OBFUSCATION_TOOLKIT ha sido reemplazado con el paquete DBMS_CRYPTO. Aun así DBMS_OBFUSCATION_TOOLKIT es todavía necesario para algunas tareas no disponibles en el paquete DBMS_CRYPTO.

TABLE_NAME
DBMS_OBFUSCATION_TOOLKIT

Comentario consulta: si aparece un valor en esta tabla debe responder "No" a la cuestión. Este valor implica que tiene acceso público y privilegios de ejecución sobre DBMS_OBFUSCATION_TOOLKIT.

No. Para remediar esta situación debe ejecutar: REVOKE EXECUTE ON DBMS_OBFUSCATION_TOOLKIT TO PUBLIC.

83.- ¿Ha establecido cifrado a nivel de Tablespace?

Comentario: cuando una tabla contiene un gran número de columnas con información de identificación personal, podría ser beneficioso cifrar la totalidad del tablespace con la finalidad de proteger su contenido.

No. Debe utilizar "tablespace encription" para proteger el contenido de aquellos tablespaces relacionados con tablas que contengan información de identificación personal.

84.- ¿Ha verificado y establecido permisos sobre el fichero radius.key?

Comentario: los permisos sobre el fichero deben estar restringidos al propietario del software de Oracle y al grupo dba. Establezca los permisos correctos sobre fichero $ORACLE_HOME/network/security/radius.key.

No. Para solucionar esta circunstancia en un sistema UNIX ejecute: chmod 440 \$ORACLE_HOME/network/security/radius.key.

85.- ¿Ha establecido el valor del parámetro SSL_CERT_REVOCATION al valor required en el fichero sqlnet.ora?

Comentario: debe asegurarse de que se requiere revocación en la comprobación de las listas de revocación de certificados utilizadas para verificar la autenticidad de los certificados del cliente.

No. Para establecer el valor del parámetro SSL_CERT_REVOCATION a required debe editar el fichero sqlnet.ora, localizar el parámetro SSL_CERT_REVOCATION y asignar el valor required.

86.- ¿Ha establecido el valor del parámetro SSL_SERVER_DN_MATCH al valor yes en el fichero sqlnet.ora?

Comentario: debe asegurarse de que la cadena DN del certificado coincide con el valor esperado.

No. Para solucionarlo modifique el valor del parámetro SSL_SERVER_DN_MATCH a yes en el fichero sqlnet.ora.

6 Arranque y parada

87.- ¿Ha comprobado que la cola está vacía antes de llevar a cabo la parada de Oracle?

Comentario: la información de cola puede accederse por encima de Oracle, más allá del control de los parámetros de seguridad. Debe establecer las mismas precauciones de seguridad utilizadas en el resto de tablas.

No. para solucionarlo ejecute: "DBMS_AQADM.PURGE_QUEUE_TABLE(queue_table=>'name.obj_qtab', purge_condition=>NULL, purge_options =>po)".

88.- ¿Ha comprobado que la cache ha sido vaciada en la parada de Oracle?

Comentario: la información contenida en la caché podría ser accedida por encima de Oracle y más allá del control de los parámetros de seguridad.

No. Para solucionarlo ejecute: "ALTER SYSTEM FLUSH BUFFER_CACHE".

7 Backup y recuperación ante desastres

89.- ¿Ha montado los ficheros de redo logs en espejo?

Comentario: la redundancia de los ficheros de redo log puede prevenir pérdidas catastróficas en el caso del fallo de un disco.

No. Establezca Mirror on-line sobre los ficheros de redo log y asegúrese de que existe más de un grupo.

90.- ¿Dispone de múltiples copias de los ficheros "control files" en varios discos físicos?

Comentario: la redundancia en los ficheros "control files" puede prevenir sobre pérdidas catastróficas en el caso de que se produzca un fallo en un disco.

No. Para remediarlo almacene los ficheros "control files" utilizando un sistema redundante como por ejemplo algún tipo de RAID.

91.- ¿Se ha asegurado de que tiene suficiente espacio para utilizar el modo Archive logs?

Comentario: si no dispone del espacio adecuado para utilizar los Archive logs el sistema podría bloquearse resultando en una denegación de servicio.

No. Localice más espacio para almacenar los ficheros que se generan cuando utiliza el modo Archive log.

92.- ¿Dispone de múltiples copias de los ficheros de redo logs distribuidas en varios discos físicos?

Comentario: la redundancia en los ficheros de redo log puede prevenir ante pérdidas catastróficas en el caso de que se produzca un fallo en un disco físico.

Sí. Dispone de múltiples copias de los ficheros de redo logs distribuidas en varios discos físicos.

93.- ¿Está realizando copias de seguridad de los ficheros de Archive log?

Comentario: los ficheros de Archive log contienen información sensible y deben ser manejados adecuadamente.

Sí. Está realizando copias de los ficheros de Archive log.

94.- ¿Está realizando verificaciones sobre los backup que se están llevando a cabo?

Comentario: los Backups realizados deben ser verificados llevando a cabo recuperaciones para asegurar que estos Backups funcionan correctamente.

Sí. Está realizando verificaciones periódicas sobre los backups realizados de tal forma que asegura que sus backups funcionan.

95.- ¿Tiene activado Oracle Failsafe?

Comentario: Oracle Failsafe utiliza la interfaz cluster server para proporcionar protección ante fallos anteriormente suministrada a través de interfaces hardware.

No. Debe activar Oracle Failsafe.

8 Parametrización de puesta a punto de perfiles

96.- ¿Ha establecido el parámetro failed_login_attempts al valor 3?

Comentario: debe restringir el número de intentos de entrada puesto que le permitirá detectar posibles ataques por fuerza bruta.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	FAILED_LOGIN_ATTEMPTS	10
MONITORING_PROFILE	FAILED_LOGIN_ATTEMPTS	UNLIMITED

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro FAILED_LOGGIN_ATTEMPTS.

No. Para remediarlo debe ejecutar la siguiente sentencia: ALTER PROFILE profile_name LIMIT_FAILED_LOGIN_ATTEMPTS 3 PASSWORD_LOCK_TIME 1.

97.- ¿Ha establecido el valor del parámetro password_life_time a valor 90 días?

Comentario: limitar el tiempo de vida de las password permitirá detectar ataques de fuerza bruta contra las cuentas de usuario.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PASSWORD_LIFE_TIME	180
MONITORING_PROFILE	PASSWORD_LIFE_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_LIFE_TIME.

No. Para remediar esta situación ejecute la siguiente sentencia: ALTER PROFILE <profile_name> LIMIT pasword_life_time 90.

98.- ¿Ha establecido el parámetro password_reuse_max a valor 20?

Comentario: este parámetro determina el número de password diferentes que pueden ser rotadas por el usuario antes de que la password actual pueda ser reutilizada.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PASSWORD_REUSE_MAX	UNLIMITED
MONITORING_PROFILE	PASSWORD_REUSE_MAX	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_REUSE_MAX.

No. Para remediar esta situación ejecute la siguiente sentencia: ALTER PROFILE <profile_name> LIMIT password_reuse_max 20.

99.- ¿Ha establecido el parámetro password_reuse_time a valor 365?

Comentario: este parámetro establece el intervalo de tiempo antes de que una password pueda ser reutilizada. Crear una ventana de tiempo amplia protegerá contra ataques de fuerza bruta.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PASSWORD_REUSE_TIME	UNLIMITED
MONITORING_PROFILE	PASSWORD_REUSE_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_REUSE_TIME.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT password_reuse_time 365.

100.- ¿Ha establecido el parámetro password_lock_time a valor 1?

Comentario: este parámetro especifica el intervalo de tiempo días que la cuenta permanecerá bloqueada si se ha alcanzado el máximo número de intentos de autenticación.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PASSWORD_LOCK_TIME	1
MONITORING_PROFILE	PASSWORD_LOCK_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_LOCK_TIME.

Sí. El valor de este parámetro permitirá que haya un intervalo de tiempo entre intentos de acceso lo que ralentizará de forma notable los ataques por fuerza bruta.

101.- ¿Ha establecido el parámetro password_grace_time a valor 3?

Comentario: este parámetro establece, en días, el intervalo de tiempo en el que se avisará al usuario para que cambie su password antes de que la password expire.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PASSWORD_GRACE_TIME	7
MONITORING_PROFILE	PASSWORD_GRACE_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_GRACE_TIME.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT password_grace_time 3.

102.- ¿Ha revisado las cuentas cuyo PASSWORD esté definido como EXTERNAL?

Comentario: debe comprobar y revisar cualquier usuario que tenga la password definida como "EXTERNAL". No debe permitir la autentificación remota de Sistema Operativo a la base de datos.

USERNAME

Comentario consulta: esta sentencia identificará aquellos usuarios cuyo password esté definido como "EXTERNAL".

Sí. Una vez revisadas no se debe permitir el uso de cuentas que tengan definido el PASSWORD como EXTERNAL.

103.- ¿Ha establecido una función de verificación a través del parámetro password_verify_function?

Comentario: debe habilitar las llamadas a password_verification_function cuando las password sean cambiadas. Esto siempre funciona cuando las password se cambian a través del comando "password".

PROFILE	RESOURCE_NAME
DEFAULT	PASSWORD_VERIFY_FUNCTION
MONITORING_PROFILE	PASSWORD_VERIFY_FUNCTION

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PASSWORD_VERIFY_FUNCTION.

No. Oracle dispone del script "utlpwdmg.sql" que puede ser utilizado para crear una función de verificación de password.

104.- ¿Ha establecido un valor adecuado en el parámetro CPU_PER_SESSION?

Comentario: permitir a un determinado usuario o aplicación consumir excesivos recursos de CPU puede degenerar en una denegación de servicio de la base de datos Oracle.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	CPU_PER_SESSION	UNLIMITED
MONITORING_PROFILE	CPU_PER_SESSION	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro CPU_PER_SESSION.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT CPU_PER_SESSION <value>.

105.- ¿Ha establecido un valor adecuado en el parámetro PRIVATE_SGA?

Comentario: permitir que una única aplicación o usuario consuma una cantidad excesiva de SGA puede resultar en una denegación de servicio de la base de datos Oracle.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	PRIVATE_SGA	UNLIMITED
MONITORING_PROFILE	PRIVATE_SGA	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro PRIVATE_SGA.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT PRIVATE_SGA <value>.

106.- ¿Ha establecido un valor adecuado en el parámetro LOGICAL_READS_PER_SESSION?

Comentario: permitir que una única aplicación o usuario lleva a cabo una gran cantidad de lecturas sobre el disco puede resultar en una denegación de servicio de la base de datos Oracle.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	LOGICAL_READS_PER_SESSION	UNLIMITED
MONITORING_PROFILE	LOGICAL_READS_PER_SESSION	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro LOGICAL_READS_PER_SESSION.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT PRIVATE_SGA <value>.

107.- ¿Ha establecido un valor adecuado en el parámetro SESSIONS_PER_USER?

Comentario: permitir que una única aplicación o usuario un número ilimitado de sesiones puede resultar en una denegación de servicio de la base de datos Oracle.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	SESSIONS_PER_USER	UNLIMITED
MONITORING_PROFILE	SESSIONS_PER_USER	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro SESSIONS_PER_USER.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT SESSIONS_PER_USER <value>.

108.- ¿Ha establecido un valor adecuado en el parámetro CONNECT_TIME?

Comentario: las sesiones abiertas durante largos periodos de tiempo pueden consumir recursos del sistema y causar una denegación de servicio para otros usuarios de la base de datos.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	CONNECT_TIME	UNLIMITED
MONITORING_PROFILE	CONNECT_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro CONNECT_TIME.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT CONNECT_TIME <value>.

109.- ¿Ha establecido un valor adecuado en el parámetro IDLE_TIME?

Comentario: las sesiones inactivas durante largos periodos de tiempo pueden consumir recursos de sistema y causar denegaciones de servicio de la base de datos Oracle.

PROFILE	RESOURCE_NAME	LIMIT
DEFAULT	IDLE_TIME	UNLIMITED
MONITORING_PROFILE	IDLE_TIME	DEFAULT

Comentario consulta: esta consulta permite identificar los perfiles, el nombre del recurso y el límite establecido asociado al parámetro IDLE_TIME.

No. Para modificar el valor de este parámetro debe ejecutar: ALTER PROFILE <profile_name> LIMIT IDLE_TIME <value>.

9 Parametrización de acceso al perfil de usuario

110.- ¿Se ha asegurado de no tener configurado el tablespace por defecto a SYSTEM para las cuentas de usuario?

Comentario: solamente el usuario SYS debería tener como Tablespace por defecto SYSTEM. Esto evita que usuarios administradores modifiquen objetos de sistema.

USERNAME	DEFAULT_TABLESPACE
SYS	SYSTEM
SYSTEM	SYSTEM
DBSNMP	SYSAUX
SYSMAN	SYSAUX
AAS11	AAS11_TABLAS
MGMT_VIEW	SYSTEM
OUTLN	SYSTEM
FLOWS_FILES	SYSAUX
MDSYS	SYSAUX
ORDSYS	SYSAUX
EXFSYS	SYSAUX
WMSYS	SYSAUX
APPQOSSYS	SYSAUX
APEX_030200	SYSAUX
OWBSYS_AUDIT	SYSAUX
ORDDATA	SYSAUX
CTXSYS	SYSAUX
ANONYMOUS	SYSAUX
XDB	SYSAUX
ORDPLUGINS	SYSAUX
OWBSYS	SYSAUX
SI_INFORMTN_SCHEMA	SYSAUX
OLAPSYS	SYSAUX
SCOTT	USERS
ORACLE_OCM	USERS
XS$NULL	USERS
BI	USERS
PM	USERS
MDDATA	USERS
IX	USERS
SH	USERS
DIP	USERS
OE	USERS
APEX_PUBLIC_USER	USERS
HR	USERS
SPATIAL_CSW_ADMIN_USR	USERS
SPATIAL_WFS_ADMIN_USR	USERS

Comentario consulta: esta consulta suministra los Tablespaces por defecto definidos para cada usuario.

No. Para solucionarlo ejecute la siguiente sentencia para los usuarios implicados: ALTER USER DEFAULT TABLESPACE table.

111.- ¿Se ha asegurado de que los usuarios de aplicación no se han concedido cuotas en los Tablespaces?

Comentario: debe establecer cuotas para los desarrolladores en los entornos compartidos de desarrollo/producción para evitar problemas de acceso a recursos.

USERNAME	TABLESPACE_NAME
AAS11	AAS11_TABLAS
OLAPSYS	SYSAUX
SYSMAN	SYSAUX
APPQOSSYS	SYSAUX
FLOWS_FILES	SYSAUX

Comentario consulta: esta consulta devuelve el nombre de usuario y el nombre del tablespace de la tabla DBA_TS_QUOTAS

No. Para remediar esta situación debe aplicar sobre los usuarios encontrados la siguiente sentencia: ALTER USER <USER_NAME> QUOTA <VALUE> ON <TABLESPACE_NAME>.

112.- ¿Ha restringido el acceso a SYS.AUD$?

Comentario: compruebe los accesos y restricciones asociados a todas las cuentas. Esto es sólo posible si el parámetro audit trail tiene el valor db o db_extended.

GRANTEE	PRIVILEGE
DELETE_CATALOG_ROLE	DELETE

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la tabla AUD$.

No.Elimine el privilegio de acceso sobre los usuarios que dispongan del mismo utilizando la siguiente sentencia: REVOKE ALL ON SYS.AUD$ FROM <USER>.

113.- ¿Ha restringido el acceso a SYS.LINK$?

Comentario: datos sensibles y las password de los usuarios son almacenadas en la tabla LINK$. Los usuarios "no Administradores" o los usuarios de sistema deben tener limitado el acceso a esta tabla.

GRANTEE	PRIVILEGE

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la tabla LINK$.

Sí. Ha comprobado el acceso a la tabla LINK$ y ha eliminado los privilegios de acceso de aquellos usuarios que no los necesitan.

114.- ¿Ha restringido el acceso a SYS.USER$?

Comentario: datos sensibles y las password de los usuarios son almacenadas en la tabla USER$. Solamente los usuarios Administradores o los usuarios de sistema deben tener acceso a esta tabla.

GRANTEE	PRIVILEGE
CTXSYS	SELECT

XDB	SELECT
APEX_030200	SELECT

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la tabla USER$.

No. Elimine todos los privilegios de acceso sobre esta tabla para aquellos usuarios que no los necesiten: REVOKE ALL ON SYS.USER$ FROM <USER>.

115.- ¿Ha restringido el acceso a SYS.SOURCE$?

Comentario: permitir a los usuarios alterar códigos sobre la tabla SOURCE$ puede comprometer la seguridad y la integridad. Compruebe cualquier cuenta que tenga acceso y limítelo siempre que sea posible.

GRANTEE	PRIVILEGE

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la tabla SOURCE$.

No. Elimine todos los privilegios de acceso sobre esta tabla para aquellos usuarios que no los necesiten: REVOKE ALL ON SYS.SOURCE$ FROM <USER>.

116.- ¿Ha restringido el acceso a cualquier tabla X$?

Comentario: las tablas X$ son utilizadas internamente por Oracle y no deben ser accedidas por parte de los usuarios. Compruebe cualquier cuenta que tenga acceso y limítelo siempre que sea posible.

GRANTEE	PRIVILEGE	TABLE_NAME
DBA	FLASHBACK	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	FLASHBACK	X$PT4O5GVJW67C95OB541F393D17GP
DBA	DEBUG	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	DEBUG	X$PT4O5GVJW67C95OB541F393D17GP
DBA	QUERY REWRITE	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	QUERY REWRITE	X$PT4O5GVJW67C95OB541F393D17GP
DBA	ON COMMIT REFRESH	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	ON COMMIT REFRESH	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	REFERENCES	X$PT4O5GVJW67C95OB541F393D17GP
DBA	UPDATE	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	UPDATE	X$PT4O5GVJW67C95OB541F393D17GP
DBA	SELECT	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	SELECT	X$PT4O5GVJW67C95OB541F393D17GP
DBA	INSERT	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	INSERT	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	INDEX	X$PT4O5GVJW67C95OB541F393D17GP
DBA	DELETE	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	DELETE	X$PT4O5GVJW67C95OB541F393D17GP
DBA	ALTER	X$PT4O5GVJW67C95OB541F393D17GP
SYSTEM	ALTER	X$PT4O5GVJW67C95OB541F393D17GP

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre las tablas X$.

No. Elimine todos los privilegios de acceso sobre estas tablas para aquellos usuarios que no los necesiten: REVOKE ALL ON X$<TABLENAME> FROM <USER>.

117.- ¿Ha restringido el acceso a cualquier vista DBA_?

Comentario: las vistas DBA suministran información sobre todos los objetos y deberían ser solamente accesibles por los administradores.

GRANTEE	PRIVILEGE	TABLE_NAME

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre las vistas DBA_$.

No. Elimine todos los privilegios de acceso sobre estas tablas para aquellos usuarios que no los necesiten: REVOKE ALL ON DBA_<TABLENAME> FROM <USER>.

118.- ¿Ha restringido el acceso a cualquier vista DBA_ROLES?

Comentario: permitir a los usuarios alterar la vista DBA_ROLES podría derivar en una escalada de privilegios e inestabilidad del sistema. Esta vista debe estar restringida excepto para los usuarios SYS y DBAs.

GRANTEE	PRIVILEGE	TABLE_NAME
SELECT_CATALOG_ROLE	SELECT	DBA_ROLES
CTXSYS	SELECT	DBA_ROLES
OLAP_XS_ADMIN	SELECT	DBA_ROLES
OLAPSYS	SELECT	DBA_ROLES

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la vista DBA_ROLES.

No. Elimine todos los privilegios de acceso sobre esta vista para aquellos usuarios que no los necesiten: REVOKE ALL ON DBA_ROLES FROM <USER>.

119.- ¿Ha restringido el acceso sobre la tabla DBA_SYS_PRIV?

Comentario: permitir el acceso a la tabla dba_sys_privs podría utilizarse para mostrar los privilegios de todos los usuarios de la bbdd. Esta tabla debe estar restringida excepto para los usuarios SYS y DBAs.

GRANTEE	PRIVILEGE	TABLE_NAME
SELECT_CATALOG_ROLE	SELECT	DBA_SYS_PRIVS
CTXSYS	SELECT	DBA_SYS_PRIVS
APEX_030200	SELECT	DBA_SYS_PRIVS

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la tabla DBA_SYS_PRIVS.

No. Elimine todos los privilegios de acceso sobre esta tabla para aquellos usuarios que no los necesiten: REVOKE ALL ON DBA_SYS_PRIVS FROM <USER>.

120.- ¿Ha restringido el acceso sobre la vista DBA_USERS?

Comentario: permitir el acceso a la vista dba_users podría utilizarse para mostrar los privilegios de todos los usuarios de la bbdd. Esta vista debe estar restringida excepto para los usuarios SYS y DBAs.

GRANTEE	PRIVILEGE	TABLE_NAME
SELECT_CATALOG_ROLE	SELECT	DBA_USERS
CTXSYS	SELECT	DBA_USERS
OLAPSYS	SELECT	DBA_USERS
APEX_030200	SELECT	DBA_USERS

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la vista DBA_USERS.

No. Elimine todos los privilegios de acceso sobre esta tabla para aquellos usuarios que no los necesiten: REVOKE ALL ON DBA_USERS FROM <USER>.

121.- ¿Ha restringido el acceso sobre la vista ROLE_ROLE_PRIVS?

Comentario: permitir el acceso a la vista dba_role_privs podría utilizarse para mostrar las concesiones de roles utilizados en la bbdd. Esta vista debe estar restringida excepto para los usuarios SYS y DBAs.

GRANTEE	PRIVILEGE	TABLE_NAME
PUBLIC	SELECT	ROLE_ROLE_PRIVS

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre la vista ROLE_ROLE_PRIVS.

No. Elimine todos los privilegios de acceso sobre esta tabla para aquellos usuarios que no los necesiten: REVOKE ALL ON ROLE_ROLE_PRIVS FROM <USER>.

122.- ¿Ha previsto la asignación de roles que contienen _CATALOG_?

Comentario: debe eliminar cualquier role de catalogo de aquellos roles y usuarios que no los necesiten.

GRANTEE	PRIVILEGE	TABLE_NAME
SELECT_CATALOG_ROLE	SELECT	SYSCATALOG_
PUBLIC	SELECT	ALL$OLAP2UCATALOGS
PUBLIC	SELECT	ALL$OLAP2UCATALOG_ENTITY_USES
PUBLIC	SELECT	ALL$OLAP2_AW_CATALOGS
PUBLIC	SELECT	ALL$OLAP2_AW_CATALOG_MEASURES

PUBLIC	SELECT	ALL$OLAP2_CATALOGS
PUBLIC	SELECT	ALL$OLAP2_CATALOG_ENTITY_USES
SELECT_CATALOG_ROLE	SELECT	DBA$OLAP2UCATALOGS
SELECT_CATALOG_ROLE	SELECT	DBA$OLAP2UCATALOG_ENTITY_USES
PUBLIC	SELECT	ALL$OLAP_CATALOGS
PUBLIC	SELECT	ALL$OLAP_CATALOG_ENTITY_USES
SELECT_CATALOG_ROLE	SELECT	DBA$OLAP_CATALOGS
SELECT_CATALOG_ROLE	SELECT	DBA$OLAP_CATALOG_ENTITY_USES
PUBLIC	SELECT	MRV$OLAP2_CATALOGS
PUBLIC	SELECT	MRV$OLAP2_CATALOG_ENTITY_USES
MGMT_USER	SELECT	MGMT$ESA_CATALOG_REPORT

Comentario consulta: esta consulta proporciona los privilegios concedidos sobre las tablas del tipo _CATALOG_.

No. Elimine la asignación de ROLES para los usuarios que no los necesiten: REVOKE ALL <ROLE> FROM <USER>.

123.- ¿Se ha asegurado cuando elimina sinónimos de que los privilegios concedidos a los sinónimos, si no son requeridos, deben ser eliminados de los objetos base?

Comentario: conceder privilegios a los sinónimos adicionalmente concede estos privilegios a los objetos de base utilizados en el sinónimo.

No. Debe asegurarse que los privilegios concedidos a los objetos base son eliminados cuando los sinónimos han sido suprimidos.

124.- ¿Ha limitado o denegado el acceso sobre el paquete de base de datos DBMS_BACKUP_RESTORE?

Comentario: este paquete proporciona al sistema de ficheros funcionalidades como la copia de ficheros, modificación de los ficheros de control, acceso a dispositivos y borrado de ficheros

GRANTEE

Comentario consulta: esta consulta suministra las concesiones realizadas sobre el paquete DBMS_BACKUP_RESTORE.

No. Para limitar o denegar el acceso a DBMS_BACKUP_RESTORE ejecute: REVOKE EXECUTE ON DBMS_BACKUP_RESTORE TO PUBLIC, REVOKE EXECTUE ON DBMS_BACKUP_RESTORE TO <USER>

125.- ¿Ha auditado la utilización del paquete de base de datos DBMS_RANDOM?

Comentario: debe auditar DBMS_RANDOM en aplicaciones para evitar usos inadecuados. No debe utilizar DBMS_RANDOM para funciones críticas relativas al cifrado o a la generación de identificadores de sesión.

GRANTEE
PUBLIC
APEX_030200

Comentario consulta: esta consulta suministra las concesiones realizadas sobre el paquete DBMS_RANDOM.

No. Para eliminar la posibilidad de utilizar el paquete DBMS_RANDOM ejecute: REVOKE EXECUTE ON DBMS_RANDOM TO PUBLIC.

126.- ¿Ha verificado los roles protegidos por password?

Comentario: los roles protegidos por password son útiles cuando una aplicación controla si un role está activo. Esto impide que un usuario acceda a la base de datos y active los privilegios asociados al role.

ROLE	PASSWORD_REQUIRED	AUTHENTICATION_TYPE
CONNECT	NO	NONE
RESOURCE	NO	NONE
DBA	NO	NONE
SELECT_CATALOG_ROLE	NO	NONE
EXECUTE_CATALOG_ROLE	NO	NONE
DELETE_CATALOG_ROLE	NO	NONE
EXP_FULL_DATABASE	NO	NONE
IMP_FULL_DATABASE	NO	NONE
LOGSTDBY_ADMINISTRATOR	NO	NONE
DBFS_ROLE	NO	NONE
AQ_ADMINISTRATOR_ROLE	NO	NONE
AQ_USER_ROLE	NO	NONE
DATAPUMP_EXP_FULL_DATABASE	NO	NONE
DATAPUMP_IMP_FULL_DATABASE	NO	NONE
ADM_PARALLEL_EXECUTE_TASK	NO	NONE
GATHER_SYSTEM_STATISTICS	NO	NONE
JAVA_DEPLOY	NO	NONE
RECOVERY_CATALOG_OWNER	NO	NONE
SCHEDULER_ADMIN	NO	NONE
HS_ADMIN_SELECT_ROLE	NO	NONE
HS_ADMIN_EXECUTE_ROLE	NO	NONE

HS_ADMIN_ROLE	NO	NONE
GLOBAL_AQ_USER_ROLE	GLOBAL	GLOBAL
OEM_ADVISOR	NO	NONE
OEM_MONITOR	NO	NONE
WM_ADMIN_ROLE	NO	NONE
JAVAUSERPRIV	NO	NONE
JAVAIDPRIV	NO	NONE
JAVASYSPRIV	NO	NONE
JAVADEBUGPRIV	NO	NONE
EJBCLIENT	NO	NONE
JMXSERVER	NO	NONE
JAVA_ADMIN	NO	NONE
CTXAPP	NO	NONE
XDBADMIN	NO	NONE
XDB_SET_INVOKER	NO	NONE
AUTHENTICATEDUSER	NO	NONE
XDB_WEBSERVICES	NO	NONE
XDB_WEBSERVICES_WITH_PUBLIC	NO	NONE
XDB_WEBSERVICES_OVER_HTTP	NO	NONE
ORDADMIN	NO	NONE
OLAPI_TRACE_USER	NO	NONE
OLAP_XS_ADMIN	NO	NONE
OWB_USER	NO	NONE
OLAP_DBA	NO	NONE
CWM_USER	NO	NONE
OLAP_USER	NO	NONE
SPATIAL_WFS_ADMIN	NO	NONE
WFS_USR_ROLE	YES	PASSWORD
SPATIAL_CSW_ADMIN	YES	PASSWORD
CSW_USR_ROLE	YES	PASSWORD
MGMT_USER	NO	NONE
APEX_ADMINISTRATOR_ROLE	NO	NONE
OWB$CLIENT	YES	PASSWORD
OWB_DESIGNCENTER_VIEW	NO	NONE

Comentario consulta: esta sentencia suministra la totalidad de características de todos los roles definidos en la base de datos. Verifique si los roles están protegidos por password.

No. Para utilizar la características de "roles identificados por password" ejecute: REVOKE ROLE <ROLE_NAME> IDENTIFIED BY <ROLE_PASSWORD>.

10 Enterprise Manager/Grid Control/Agentes

127.- ¿Ha limitado el acceso sobre la aplicación "Oracle Enterprise Management"?

Comentario: sin el establecimiento de limitaciones sobre el acceso al "Enterprise Management" el acceso a los agentes remotos es virtualmente ilimitado.

No. Debe limitar el acceso a la aplicación "Oracle Enterprise Management".

128.- ¿Está monitorizando el tamaño de las subidas realizadas desde "Enterprise Agent"?

Comentario: debe determinar si existen tamaños inusuales o se ha incrementado de forma sospechosa el tamaño de los ficheros, porque podría implicar la presencia de un agente malicioso.

No. Debería crear un monitor para controlar el tamaño de los ficheros procedentes de "Enterprise Agent". Para este fin puede utilizar el comando "status agent".

129.- ¿Está utilizando la funcionalidad "Enterprise Manager Framework Security" donde sea posible?

Comentario: "Enterprise Manager Framework security" utiliza comunicaciones seguras entre varios "Enterprise Manager Components".

No. Para solucionarlo habilite HTTPS entre los "management agents" y los "mangement services".

130.- ¿Ha configurado un valor apropiado para "Grid Control Timeout" en el servidor de aplicaciones Oracle?

Comentario: para protegerse de accesos no autorizados a "Grid Control" utilizando un navegador, debe establecer un valor adecuado de timeout. Un valor de 30 minutos o menos es el recomendado.

No. Para solucionarlo debe editar $IAS_HOME/sysman/config/emoms.properties y establecer un valor igual o inferior a 30.

131.- ¿Está evitando el uso de comandos que contienen password desde línea de comandos?

Comentario: los comandos pueden ser capturados por otros usuarios con acceso al sistema (Ejemplo: Unix utilizando el comando ps). Algunas Shells disponen de históricos de comandos de donde pueden ser recuperados.

No. Desde línea de comandos debe evitar el uso de comandos que contienen la password.

132.- ¿Dispone de una cuenta de usuario separada para la utilización de "Inteligent Agent"?

Comentario: las cuentas utilizadas para los agentes de base de datos deben estar separadas. Esto aislará a los agentes del resto del servidor Oracle.

No. En entornos Unix debe crear una cuenta exclusiva de usuario para la utilización de "Inteligent Agent". Las cuentas separadas no están recomendadas en entornos Windows.

11 Elementos relevantes para subsistemas específicos

133.- ¿Ha verificado las sugerencias propuestas por "Automatic Database Diagnostic Monitor (ADDM)"?

Comentario: las sugerencias propuestas por la herramienta "Automatic Database Diagnostic Monitor (ADDM)" no deben remplazar en ningún caso el conocimiento del DBA.

Sí. Está considerando las sugerencias propuestas por la herramienta "Automatic Database Diagnostic Monitor (ADDM)".

134.- ¿Está monitorizando la característica "Automated Memory Manager"?

Comentario: la utilización Automated Memory Manager no debe implicar dejar de utilizar un DBA para monitorizar el estado del sistema.

Sí. Está monitorizando la característica "Automated Memory Manager".

135.- ¿Está utilizando "Automatic Workload Repository" para guardar todas las estadísticas de desempeño del sistema?

Comentario: la herramienta AWR es la central de todo el framework para la gestión automática. Trabaja con elementos internos de la bbdd utilizados para la detección de problemas y para acciones de auto-tuning.

Sí. Está utilizando AWR para guardar todas las estadísticas de eficiencia del sistema. Las estadísticas son relativas a la utilización de objetos, eficiencia de sentencias SQL, históricos de sesión...

136.- ¿Está utilizando el control de acceso de grano fino dentro de los objetos?

Comentario: el control de acceso de grano fino puede proporcionar seguridad a nivel de columna y de fila. Este elemento puede suministrar una capa adicional para el control de acceso a objetos, limitándolo.

Sí. Está utilizando el control de acceso de grano fino dentro de los objetos. Para que el acceso de grano fino funcione correctamente, debe activar la optimización basada en costes.

12 Políticas generales y procedimientos

137.- ¿Ha instalado Oracle en un servidor que no esté expuesto directamente a Internet?

Comentario: Oracle debe ser únicamente instalado en un sistema protegido por un firewall u otro sistema de protección de red equivalente.

Sí. Ha instalado Oracle en un servidor que no esté expuesto directamente a Internet.

138.- ¿Revisa periódicamente el contenido del fichero alert.log?

Comentario: el fichero alert.log de Oracle debe ser regularmente revisado para detectar errores. Algunos errores pueden indicar que el sistema está bajo un ataque.

Sí. Está revisando periódicamente el fichero alert.log en busca de errores.

139.- ¿Ha borrado o a protegido el script de base de datos utilizado para su creación?

Comentario: los scripts de creación pueden suministrar a los atacantes información de valor sobre Oracle y la instancia.

Sí. Ha borrado o ha protegido el script de base de datos utilizado para su creación.

140.- ¿Ha prohibido a "Oracle" como miembro del grupo root?

Comentario: el propietario de la cuenta de software de Oracle no debe ser un miembro del grupo root en sistemas Unix.

Sí. Ha prohibido a "Oracle" como miembro del grupo root.

141.- ¿Ha revisado los miembros del grupo DBA para asegurarse de que sólo las cuentas autorizadas están incluidas?

Comentario: los miembros de este grupo únicamente deben ser aquellos que requieran privilegios de DBA.

Sí. Ha revisado los miembros del grupo DBA para asegurarse de que sólo las cuentas autorizadas están incluidas.

142.- ¿Ha evitado impedir acceder al nombre de un usuario o a su password desde la lista de procesos?

Comentario: tener accesible el nombre de usuario y la password en la lista de procesos podría permitir que alguien fuera capaz de extraer un conjunto de datos de usuario utilizados para acceder a la bbdd.

Sí. Ha evitado tener accesible el nombre de usuario y la password a partir de la información contenida en la lista de procesos.

143.- ¿Ha evitado impedir acceder al nombre de un usuario o a su password desde el planificador de tareas "cron"?

Comentario: tener accesible el nombre de usuario y la password en un trabajo planificado con "cron" podría permitir que alguien fuera capaz de extraer datos de usuario utilizables para acceder a la bbdd.

Sí. Ha evitado tener accesible el nombre de usuario y la password a partir de la información contenida en los trabajos planificados de cron.

144.- ¿Ha eliminado cualquier cuenta de desarrollador que exista en la base de datos de producción?

Comentario: debe eliminar cualquier cuenta de desarrollador que exista en la bbdd de producción para que no haya cuentas con privilegios no permitidos. Para comprobarlo, edite el fichero /etc/password.

Sí. Ha eliminado cualquier cuenta de desarrollador que exista en la base de datos de producción.

145.- ¿Ha cambiado las password de las bases de datos de desarrollo o test?

Comentario: si las bbdd's de desarrollo o test están generadas a partir de backups o exports de la bbdd de producción, las passwords deben cambiarse antes de permitir el acceso a desarrolladores o probadores.

Sí. Ha cambiado las password de las bases de datos de desarrollo o test.

146.- ¿Ha eliminado la información sensible de los entornos de desarrollo o test?

Comentario: si las bbdd's de desarrollo o test están generadas a partir de backups o exports de la bbdd de producción, debe eliminar toda la información sensible antes de permitir el acceso a estos entornos.

Sí. Ha eliminado la información sensible de los entornos de desarrollo o test.

147.- ¿Ha eliminado la posibilidad de que existan "puertas traseras", realizando un seguimiento de los cambios en los procedimientos y aplicando sumas de comprobación sobre el código fuente?

Comentario: realizar un seguimiento de los cambios en los procedimientos y aplicar sumas de comprobación en el código fuente podría evitar la existencia de "puertas traseras" que permitan el acceso a la bbdd.

Sí. Está realizando un seguimiento de los cambios en los procedimientos y aplicando sumas de comprobación sobre el código fuente

148.- ¿Ha evitado exponer la información de configuración interna?

Comentario: exponer la información de configuración interna brinda a los atacantes una lista de objetivos y líneas de actuación dentro del entorno Oracle.

Sí. Ha evitado exponer la información de configuración interna del entorno Oracle.

149.- ¿Ha configurado los protectores de pantalla con password y activables cada 15 minutos?

Comentario: los protectores de pantalla mal configurados constituyen para los atacantes una posibilidad para tomar el control de un usuario que ya ha entrado en Oracle.

Sí. Ha configurado los protectores de pantalla con password y activables cada 15 minutos.

150.- ¿Revisa periódicamente los eventos y logs de sistema?

Comentario: excesivos errores o errores "raros" pueden ser indicativos de que un sistema o bbdd está bajo un ataque. Debe revisar apropiadamente los log de sistema para mantener la integridad del sistema.

Sí. Está revisando periódicamente los eventos y logs del sistema.

151.- ¿Se ha asegurado de utilizar certificados SSL cuando utiliza "Redo Transport Services"?

Comentario: las conexiones establecidas a través de la red pueden ser monitorizadas e interceptadas exponiendo información sensible.

Sí. Se ha asegurado de utilizar certificados SSL cuando utiliza "Redo Transport Services".

152.- ¿Se ha asegurado que los "Incident Packages" han sido destruidos o adecuadamente protegidos después de ser subidos a Oracle?

Comentario: la información sensible puede estar contenida en los "Incident packages". No mantener protegidos estos elementos podría provocar la pérdida de información sensible relacionada con Oracle.

Sí. Se ha asegurado que los "Incident Packages" han sido destruidos o adecuadamente protegidos después de ser subidos a Oracle.

13 Políticas de auditoría y procedimientos

153.- ¿Ha eliminado los esquemas de base de datos que no están siendo utilizados?

Comentario: dejar esquemas adicionales en la base de datos puede proporcionar a los atacantes detalles adicionales sobre el uso del sistema Oracle o incluso información sensible.

Sí. Ha eliminado los esquemas de base de datos que no están siendo utilizados

154.- ¿Tiene auditado todos los eventos de "logons" y "logoffs"?

Comentario: auditar los eventos de "logons" y "logoffs" puede suministrar información adicional para aislar la causa de incidentes de seguridad.

USER_NAME	SUCCESS	FAILURE
	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devolverá los accesos con éxito y fallidos de los usuarios que tengan el privilegio "CREATE SESSION".

Sí. Tiene auditado todos los eventos de "logons" y "logoffs".

155.- ¿Ha auditado a través de la opción "ACCESS WHENEVER NOT SUCCESSFUL" los intentos de acceso fallidos?

Comentario: la auditoría por SESSION mostrará únicamente un evento de auditoría por cada intento de acceso. Registrar los intentos fallidos detectará cualquier sentencia que intente acceder a una tabla y falle.

OWNER	OBJECT_NAME	OBJECT_TYPE	ALT	AUD	COM	DEL	GRA	IND	INS	LOC	REN	SEL	UPD	REF	EXE	CRE	REA	WRI	FBK

Comentario consulta: esta consulta proporciona detalle de la información de acceso sobre la tabla CUESTIONES.

No. Para auditar los intentos de acceso fallidos debe ejecutar: AUDIT SELECT ON TABLE WHENEVER NOT SUCCESSFUL.

156.- ¿Tiene habilitada la acción de auditoría "Audit ALTER ANY TABLE"?

Comentario: las acciones de Alter Table "no autorizadas" pueden acarrear fallos de aplicación o formar parte de un ataque.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es ALTER_ANY_TABLE.

No. Para auditar las acciones de ALTER TABLE ejecute la siguiente sentencia: Audit ALTER ANY table.

157.- ¿Tiene habilitada la acción de auditoría "Audit ALTER USER"?

Comentario: las acciones de Alter User "no autorizadas" pueden acarrear fallos de aplicación, o formar parte de un ataque.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es ALTER_USER.

No. Para auditar las acciones de ALTER USER ejecute la siguiente sentencia: Audit ALTER USER.

158.- ¿Tiene habilitada la acción de auditoría "Audit CREATE ANY <object>"?

Comentario: auditar la creación de objetos, como tablas o bases de datos, proporcionará un registro de eventos que podría ser útil cuando se investigan problemas de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		CREATE ANY TABLE	BY ACCESS	BY ACCESS
		CREATE ANY PROCEDURE	BY ACCESS	BY ACCESS
		CREATE EXTERNAL JOB	BY ACCESS	BY ACCESS
		CREATE ANY JOB	BY ACCESS	BY ACCESS
		CREATE ANY LIBRARY	BY ACCESS	BY ACCESS
		CREATE PUBLIC DATABASE LINK	BY ACCESS	BY ACCESS
		CREATE USER	BY ACCESS	BY ACCESS
		CREATE SESSION	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es del tipo CREATE.

No. Para auditar las acciones de ALTER CREATE ANY ejecute la siguiente sentencia: Audit CREATE ANY <object>.

159.- ¿Tiene habilitada la acción de auditoría "Audit CREATE ROLE"?

Comentario: auditar la creación de roles proporcionará un registro para asegurar el uso de privilegios en la administración de las cuentas. Es útil cuando se investigan ciertos eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es CREATE ROLE.

No. Para auditar las acciones de CREATE ROLE ejecute la siguiente sentencia: Audit CREATE ROLE.

160.- ¿Tiene habilitada la acción de auditoría "Audit CREATE USER"?

Comentario: auditar la creación de usuarios proporcionará un registro para asegurar el uso de privilegios en la administración de las cuentas. Es útil cuando se investigan ciertos eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		CREATE USER	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es CREATE USER.
No. Para auditar las acciones de CREATE USER ejecute la siguiente sentencia: Audit CREATE USER.

161.- ¿Tiene habilitada la acción de auditoría "Audit CREATE SESSION"?

Comentario: debe auditar el uso de CREATE SESSION para operaciones con éxito o fallidas. Esta información puede ser útil para depurar aplicaciones y fallos en las sesiones de usuario.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		CREATE SESSION	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es CREATE SESSION.
No. Para auditar las acciones de CREATE SESSION ejecute la siguiente sentencia: Audit CREATE SESSION.

162.- ¿Tiene habilitada la acción de auditoría "Audit DROP(PRIV)"?

Comentario: auditar la eliminación de objetos de la base de datos como tablas o bases de datos podría proporcionar un registro de eventos útiles cuando se investigan problemas de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		DROP ANY TABLE	BY ACCESS	BY ACCESS
		DROP ANY PROCEDURE	BY ACCESS	BY ACCESS
		DROP USER	BY ACCESS	BY ACCESS
		DROP PROFILE	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es del tipo DROP.
No. Para auditar las acciones de DROP ejecute la siguiente sentencia: Audit DROP(PRIV).

163.- ¿Tiene habilitada la acción de auditoría "Audit DROP ANY PROCEDURE"?

Comentario: debe auditar el uso de "DROP ANY PROCEDURE". Auditar la eliminación de procedimientos de la bbdd proporcionará un registro que puede ser útil cuando se investiga la ocurrencia de eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es DROP PROCEDURE.
No. Para auditar las acciones de "DROP ANY PROCEDURE" ejecute la siguiente sentencia: Audit DROP ANY PROCEDURE.

164.- ¿Tiene habilitada la acción de auditoría "Audit DROP ANY TABLE"?

Comentario: auditar la eliminación de tablas de la base de datos puede proporcionar un registro de determinadas acciones que podría ser útil cuando investigamos eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		DROP ANY TABLE	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es DROP ANY TABLE.
No. Para auditar las acciones de "DROP ANY TABLE" ejecute la siguiente sentencia: Audit DROP ANY TABLE.

165.- ¿Tiene habilitada la acción de auditoría "Audit GRANT ANY ROLE"?

Comentario: auditar la concesión de privilegios proporciona un registro para asegurar el uso adecuado de privilegios en la administración de cuentas. Esto podría ser útil cuando investigamos eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		GRANT ANY ROLE	BY ACCESS	BY ACCESS

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es GRANT ANY ROLE.
No. Para auditar las acciones de "GRANT ANY ROLE" ejecute la siguiente sentencia: Audit GRANT ANY ROLE.

166.- ¿Tiene habilitada la acción de auditoría "Audit INSERT failures"?

Comentario: auditar fallos en los insert puede ser útil cuando se investigan ciertos eventos de seguridad cómo intentos de inyección de código SQL.

OBJECT_NAME	INS

Comentario consulta: esta consulta proporciona los campos OBJECT_NAME y el INS de la tabla DBA_STMT_AUDIT_OPTS.

No. Para auditar las acciones de "Audit INSERT failures" ejecute la siguiente sentencia: AUDIT INSERT ON objectname WHENEVER NOT SUCCESSFUL.

167.- ¿Tiene habilitada la acción de auditoría "Audit EXECUTE PROCEDURE"?

Comentario: auditar "EXECUTE PROCEDURE" proporcionará un registro de los procedimientos que fueron ejecutados y por quién. Esta información puede ser útil cuando se investigan ciertos eventos de seguridad.

USER_NAME	PROXY_NAME	AUDIT_OPTION	SUCCESS	FAILURE
		EXECUTE PROCEDURE	BY SESSION	BY SESSION

Comentario consulta: esta consulta devuelve la totalidad de datos de la tabla DBA_STMT_AUDIT_OPTS donde la opción de auditoría es EXECUTE PROCEDURE.

No. Para auditar las acciones de "Audit EXECUTE PROCEDURE" ejecute la siguiente sentencia: AUDIT EXECUTE PROCEDURE.

14 Apéndice A. Configuración adicional

168.- ¿Está utilizando Oracle Label Security cuando es posible?

Comentario: OLS es una capa adicional utilizada para la seguridad que puede usarse para la creación de una Virtual Private Database (VPD).

No. Siempre que sea posible debe habilitar y aplicar Oracle label Security.

169.- ¿Está ocultando las "label column" cuando utiliza Oracle Label Security?

Comentario: siempre que sea posible, en el caso de utilizar OLS, debe ocultar las "label column".

No. Puede ocultarlas pasando la directiva HIDE al parámetro DEFAULT_OPTIONS para SA_SYSDBA.CREATE_POLICY(global) o el parámetro TABLE_OPTIONS para el procedimiento APPLY_TABLE_POLICY(una tabla).

170.- ¿Ha incluido LABEL_UPDATE cuando utiliza Oracle Label Security?

Comentario: utilizar LABEL_UPDATE asegura que el usuario no puede reclasificar los datos almacenados cambiando las etiquetas.

No. Debe incluir LABEL_UPDATE cómo valor para el parámetro TABLE_OPTIONS cuando aplica la política de Oracle Label Security sobre una tabla.

171.- ¿Ha limitado las posibles manipulaciones cuando utiliza Oracle Label Security?

Comentario: mediante la creación de un procedimiento, es evitada la manipulación directa por parte de los usuarios de las etiquetas.

No. Siempre que sea posible debe utilizar un procedimiento de confianza para limitar y controlar la manipulación de etiquetas.

172.- ¿Ha realizado un backup de los datos antes de utilizar Oracle Label Security?

Comentario: utilizar OLS añade nuevas columnas o las oculta lo que puede hacer la tabla inutilizable. Para aplicaciones que utilizan todos los datos estás columnas pueden interpretarse como datos corruptos.

No. Realice un backup de los datos antes de aplicar Oracle Label Security.

173.- ¿ha almacenado las etiquetas en Oracle Internet Directory (OID) siempre que ha sido posible?

Comentario: dentro del contexto de la utilización de Enterprise User Security option, se dispondrá de un método de gestión centralizado para password de usuario, roles y autorizaciones OLS.

No. Siempre que sea aplicable almacene las etiquetas en el Oracle Internet Directory (OID).

174.- ¿El sistema de ficheros que utiliza ha sido montado utilizando volúmenes en RAID?

Comentario: los sistemas de ficheros en los que reside Oracle deben estar montados sobre volúmenes en RAID.

No. Debe crear particiones en RAID para la base de datos Oracle y los ficheros.

175.- ¿Limpia magnéticamente los discos que han fallado?

Comentario: los datos sensibles o la información, pueden ser recuperados de aquellos medios que no han sido correctamente borrados.

No. Debe limpiar magnéticamente los discos antiguos, no utilizados durante mucho tiempo y aquellos que han fallado. Esta acción debe ser llevada a cabo por los administradores del sistema.

176.- ¿Ha verificado los permisos sobre los backups de los discos del sistema?

Comentario: en muchos entornos, los backups de la bbdd se escriben en los discos del sistema. En este tipo de entornos, debe asegurarse que los ficheros de backup están protegidos.

No. Debe establecer los permisos apropiados sobre los ficheros de base de datos almacenados en las cintas de backup.

177.- ¿Está almacenando las copias de seguridad en un sitio diferente?

Comentario: una de las mejores prácticas es tener en sitios diferentes backups ante la posibilidad de una catástrofe física.

No. Debe llevar a cabo un protocolo para almacenar las copias de seguridad en un sitio diferente.

178.- ¿Están documentados y probados los procedimientos de recuperación?

Comentario: no tener documentados y probados los procedimientos de recuperación establecidos puede resultar en una pérdida de datos y comprometer la integridad del sistema.

No. Debe asegurarse de que los procedimientos de recuperación de la base de datos están completamente documentados y que son testeados periódicamente.

179.- ¿Tiene habilitado un "Screening router" para limitar el acceso al servidor de base de datos?

Comentario: no restringir el acceso a la base de datos extiende la superficie de ataque. El acceso a través de la red debe estar limitado a las aplicaciones y a los administradores.

No. Debe habilitar un "Screening router" para limitar el acceso al servidor de base de datos.

180.- ¿Tiene habilitado un "firewall" sobre las máquinas utilizadas para administrar la base de datos?

Comentario: debe utilizar un firewall en la totalidad de ordenadores utilizados para administrar las bases de datos de forma remota.

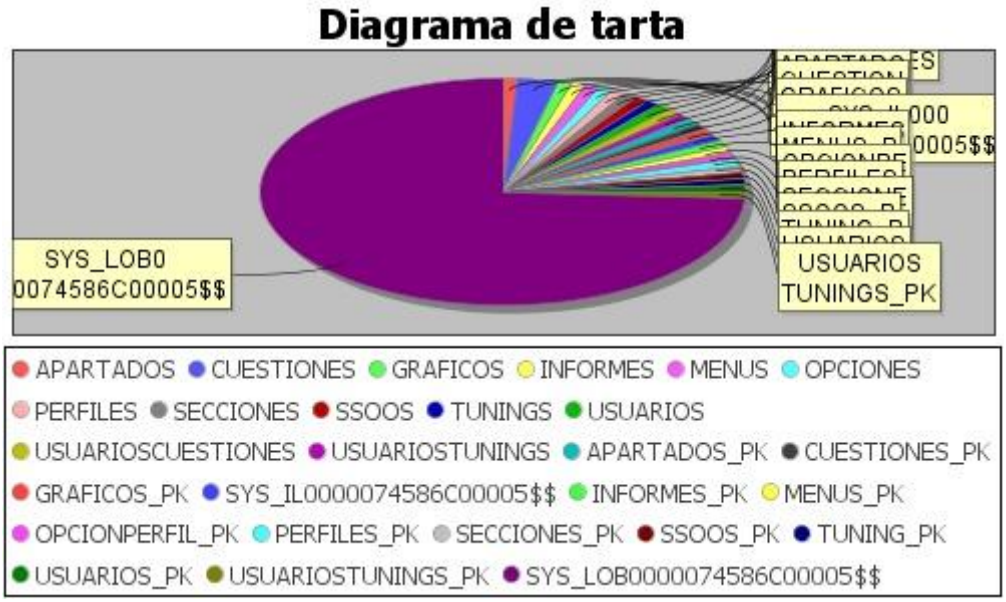
No. Debe habilitar un "firewall" sobre las máquinas utilizadas para administrar la base de datos.

1 Espacio de almacenamiento: Tablespaces,segmentos

1.- Espacio en MB ocupado por los segmentos asociados al Tablespace "AAS11_TABLAS".

Comentario: esta consulta proporciona el nombre de cada uno de los segmentos y la cantidad de MB ocupados en el Tablespace "AAS11_TABLAS".

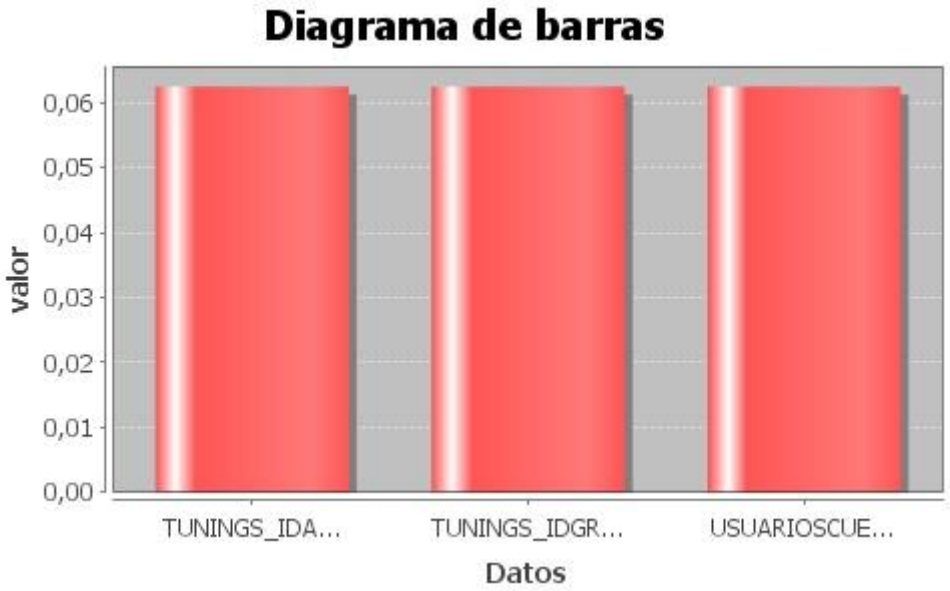
SEGMENT	MB
APARTADOS	0.0625
CUESTIONES	0.1875
GRAFICOS	0.0625
INFORMES	0.0625
MENUS	0.0625
OPCIONES	0.0625
PERFILES	0.0625
SECCIONES	0.0625
SSOOS	0.0625
TUNINGS	0.0625
USUARIOS	0.0625
USUARIOSCUESTIONES	0.0625
USUARIOSTUNINGS	0.0625
APARTADOS_PK	0.0625
CUESTIONES_PK	0.0625
GRAFICOS_PK	0.0625
SYS_IL0000074586C00005$$	0.0625
INFORMES_PK	0.0625
MENUS_PK	0.0625
OPCIONPERFIL_PK	0.0625
PERFILES_PK	0.0625
SECCIONES_PK	0.0625
SSOOS_PK	0.0625
TUNING_PK	0.0625
USUARIOS_PK	0.0625
USUARIOSTUNINGS_PK	0.0625
SYS_LOB0000074586C00005$$	5



2.- Espacio en MB ocupado por los segmentos asociados al Tablespace "AAS11_INDICES".

Comentario: esta consulta proporciona el nombre de cada uno de los segmentos y la cantidad de MB ocupados en el Tablespace "AAS11_INDICES".

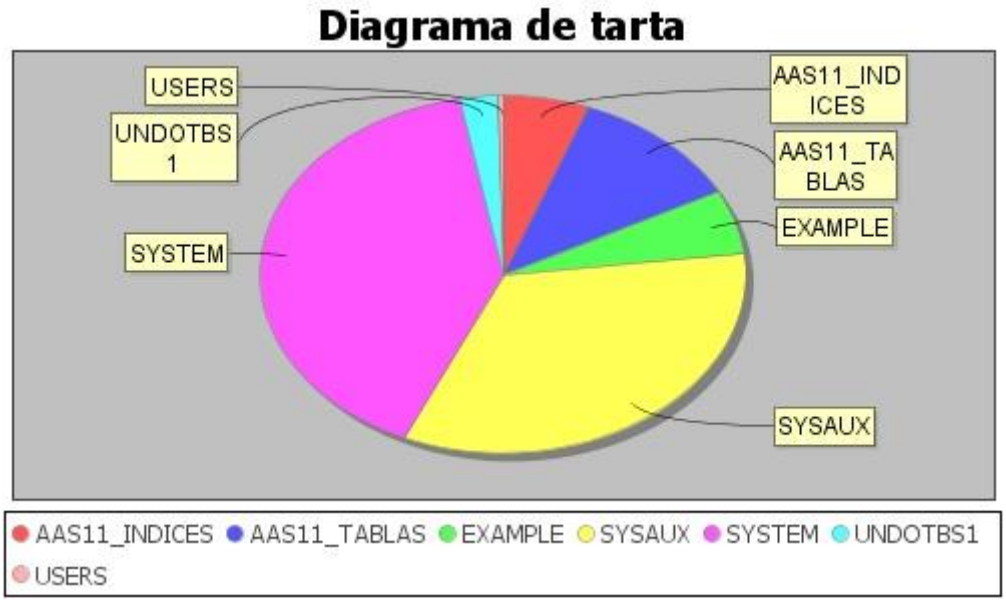
SEGMENT	MB
TUNINGS_IDAPARTADO_IDX	0.0625
TUNINGS_IDGRAFICO_IDX	0.0625
USUARIOSCUESTIONES_PK_IDX	0.0625



7.- Tamaño actual de cada uno de los Tablespaces expresado en MB ordenado por nombre de Tablespace.

Comentario: esta consulta proporciona el tamaño actual de cada uno de los Tablespaces expresado en MB ordenado por nombre de Tablespace.

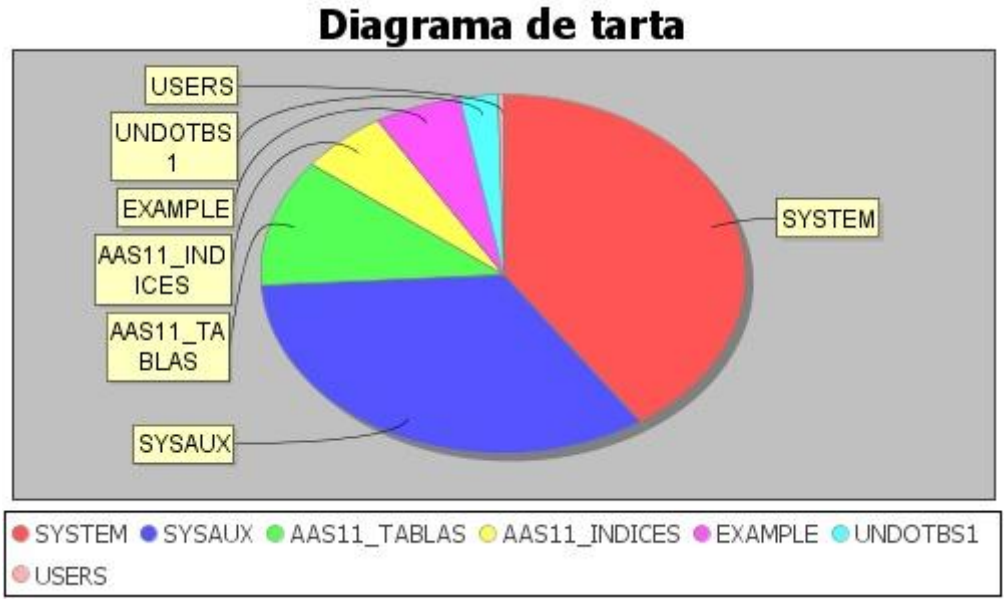
TABLESPACE	TOTAL (MB)
AAS11_INDICES	100
AAS11_TABLAS	200
EXAMPLE	100
SYSAUX	580
SYSTEM	700
UNDOTBS1	45
USERS	5



20.- Tamaño de los Tablespaces ordenado de mayor a menor.

Comentario: esta consulta suministra el tamaño de los Tablespaces ordenado de mayor a menor.

TABLESPACE_NAME	TAMANO
SYSTEM	700
SYSAUX	580
AAS11_TABLAS	200
AAS11_INDICES	100
EXAMPLE	100
UNDOTBS1	45
USERS	5



2 SGA: Distribución del almacenamiento en memoria.

3.- Tamaño del Redolog Buffer en Kilo bytes.

Comentario: esta consulta proporciona el tamaño del Redolog Buffer en Kilo bytes.

'REDOSIZE(KB)'	STATUS
Redo size (Kb)	51200

4.- Tamaño de la System Global Area en MB.

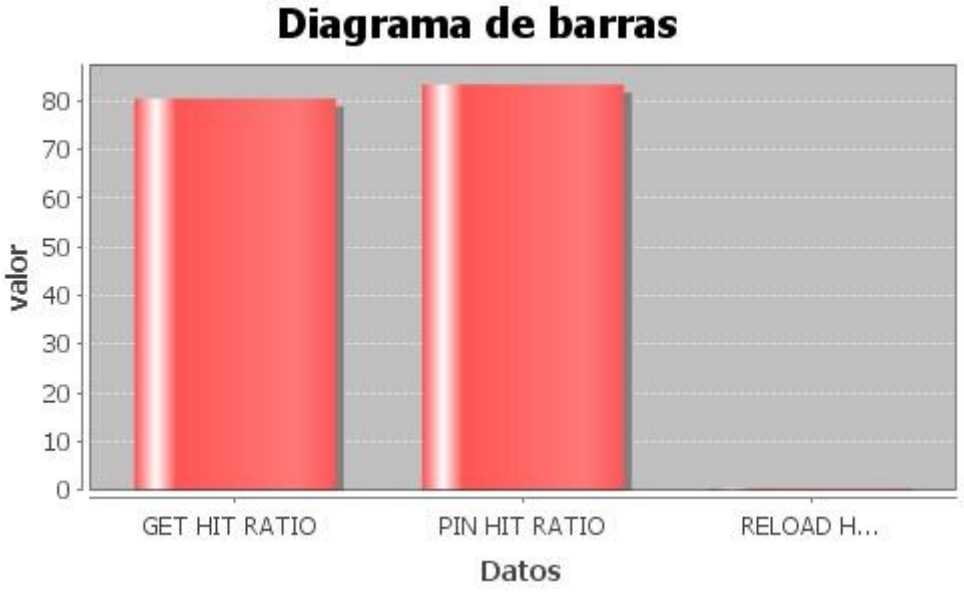
Comentario: esta consulta proporciona el tamaño de la System Global Area en MB.

'SGA(MB)'	STATUS_02
SGA (Mb)	1011

11.- Estadísticas asociadas a la library caché.

Comentario: PINS: N. de veces que un objeto de la librería fue ejecutado. RELOADS N. de pérdidas cuando se ejecutaba un objeto GETHITRATIO. GETHTS/GETS: veces que un objeto ha ejecutado estando ya parseado. PINHITS/PINS: cercano a 1 indica que la mayoría de los objetos han sido puestos en la cache. Debe Fijarse en este parámetro.

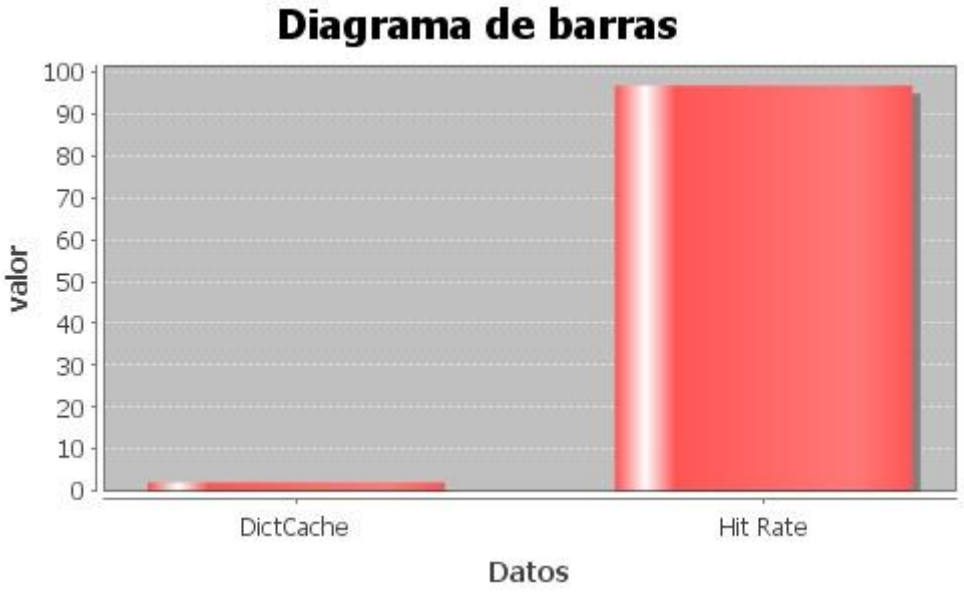
ESTADISTICO	VALOR
GET HIT RATIO	80.39
PIN HIT RATIO	83.28
RELOAD HIT RATIO	0.38



12.- Estadísticas asociadas a la Dictionary Caché.

Comentario: se debe intentar mantener el Hit Rate por encima de 90% y el DictCache por debajo de 5% para guardar el caché del diccionario de datos en la SGA. Debe aumentar la SHARED_POOL_SIZE para lograrlo.

ESTADISTICO	VALOR
DictCache	2.03
Hit Rate	96.75



13.- Información asociada a la User Global Area (UGA).

Comentario: esta consulta proporciona información asociada al User Global Area (informe de la memoria por usuario).

DETALLE	BYTES
Total de Memoria para todas las sesiones	9857488
Máxima memoria para todas las sesiones	29549536

14.- Información relativa al tamaño de la Shared Pool.

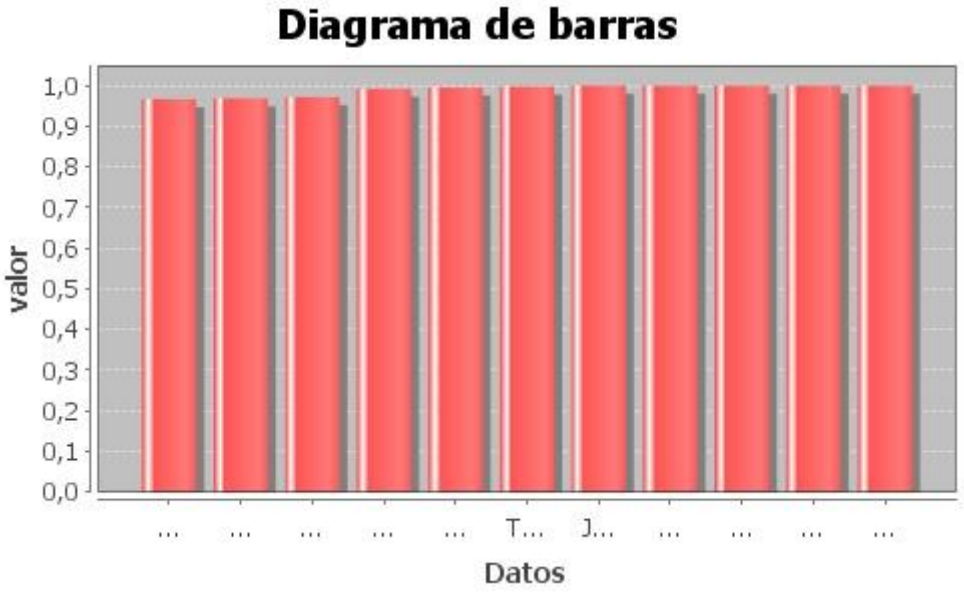
Comentario: esta consulta suministra el valor del tamaño de la Shared Pool.

POOL	SUM(BYTES)
shared pool	406852280

15.- Información de las cachés de lecturas clasificadas por sesión.

Comentario: esta consulta proporciona información de las cachés de lecturas clasificadas por sesión.

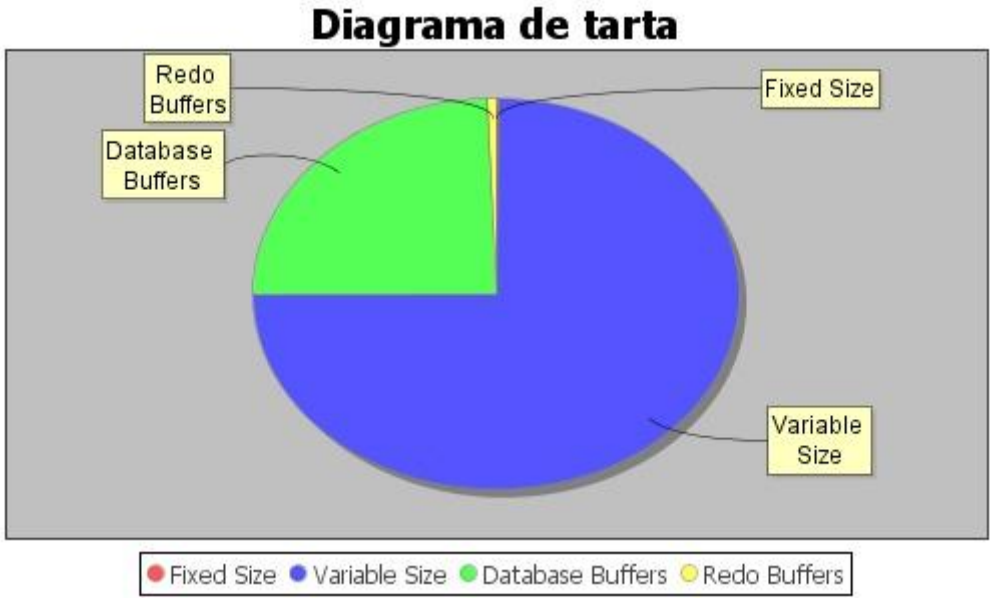
PROGRAMA	RATIO
ORACLE.EXE (DBRM)	0.967
ORACLE.EXE (MMON)	0.9689
ORACLE.EXE (SMON)	0.9722
ORACLE.EXE (RECO)	0.9929
ORACLE.EXE (CJQ0)	0.9958
Toad.exe	0.9982
JDBC Thin Client	1
ORACLE.EXE (Q000)	1
JDBC Thin Client	1
JDBC Thin Client	1
JDBC Thin Client	1
JDBC Thin Client	1
JDBC Thin Client	1
JDBC Thin Client	1
ORACLE.EXE (Q002)	1
ORACLE.EXE (QMNC)	1
ORACLE.EXE (MMNL)	1
JDBC Thin Client	1



16.- Informe de distribución de tamaños de la SGA.

Comentario: esta consulta proporciona el informe de distribución de tamaños de la SGA.

ZONA	VALOR
Fixed Size	2.08
Variable Size	756
Database Buffers	248
Redo Buffers	5.37



17.- Informe de la memoria libre clasificada por zona de memoria.

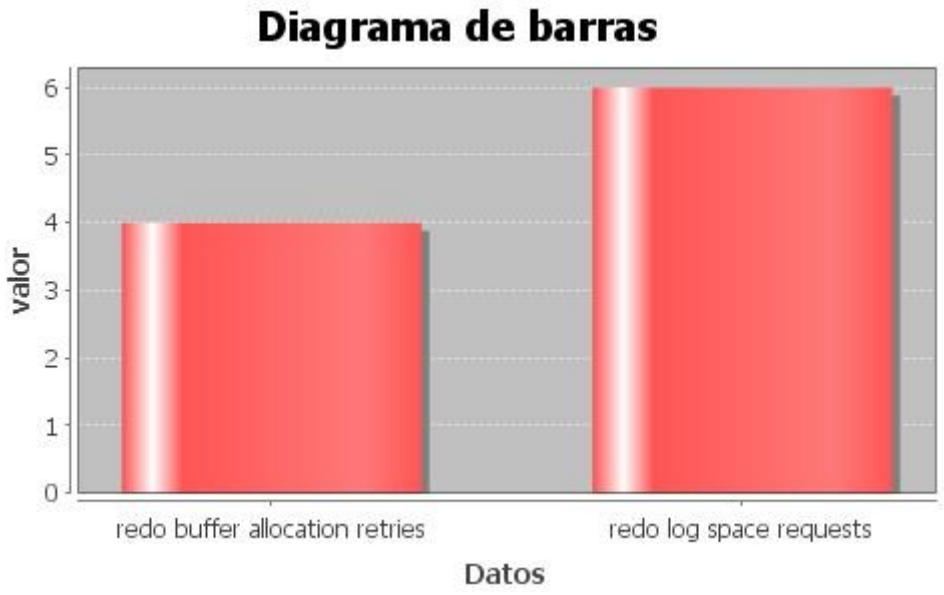
Comentario: esta consulta proporciona el informe de la memoria libre clasificada por zona de memoria.

TIPO	Memoria Libre - POOL (Mb)
free memory shared pool	54.27
free memory large pool	3.53
free memory java pool	12

18.- Estadísticas asociadas a los redo log.

Comentario: este valor debería de ser cercano a 0. Si el valor aumenta constantemente los procesos tienen que esperar para espacio libre en el REDO BUFFER. Aumentar cada vez el parámetro LOG_BUFFER un 5% hasta alcanzar un valor cercano a 0. Este parámetro está expresado en bytes y debe de ser múltiplo de DB_BLOCK_SIZE.

NAME	VALUE
redo buffer allocation retries	4
redo log space requests	6



19.- Ordenaciones en la Sort Area Size.

Comentario: ordenaciones en la Sort Area Size.

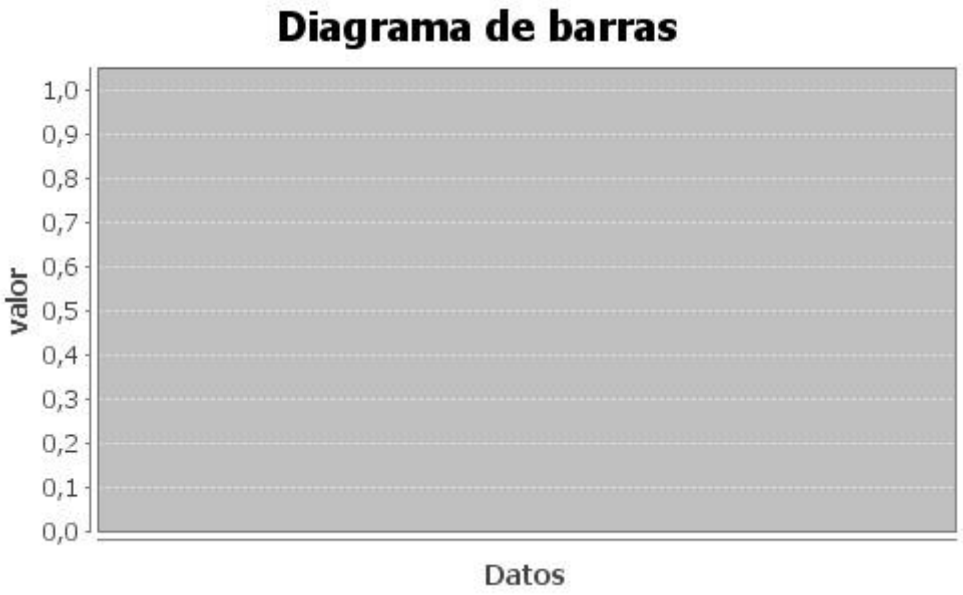
ESTADISTICO	ROUND(MEM/(MEM+DISK)*100,3)
Memory Sort Ratio	100

3 Objetos de la base de datos

5.- Número de objetos "inválidos" definidos en el sgbd clasificados por propietario.

Comentario: esta consulta proporciona el número de objetos "inválidos" definidos en el sgbd clasificados por propietario.

OBJECT_TYPE	QUANTITY



6.- Número de objetos modificados en el esquema AAS11 en el plazo de 40 días.

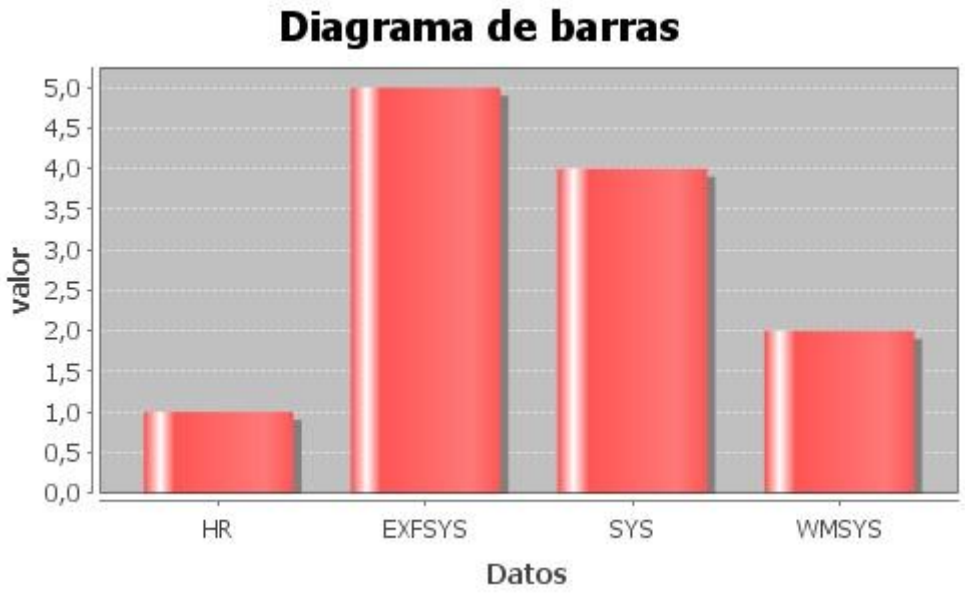
Comentario: esta consulta proporciona el número de objetos modificados en el esquema AAS11 en el plazo de 40 días.

OBJET	DAYS
AAS11.TUNINGS	34.26

9.- Número de Triggers deshabilitados clasificados por propietario.

Comentario: esta consulta suministra los Triggers deshabilitados clasificados por propietario.

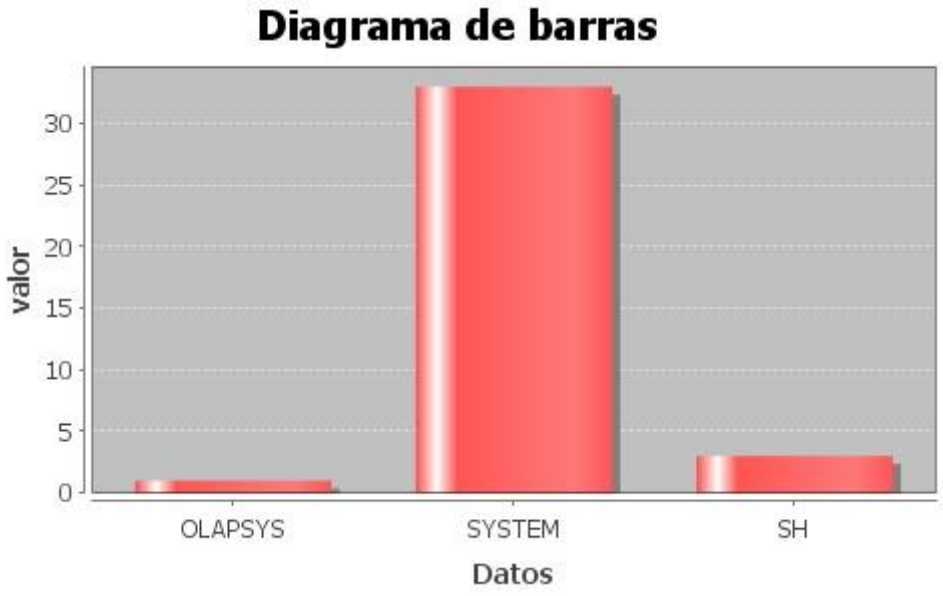
TRIGGERS DESHABILITADOS	COUNT(*)
HR	1
EXFSYS	5
SYS	4
WMSYS	2



10.- Número de Constraints deshabilitados clasificados por propietario.

Comentario: esta consulta suministra el número de Constraints deshabilitados clasificados por propietario.

CONSTRAINTS DESHABILITADAS	COUNT(*)
OLAPSYS	1
SYSTEM	33
SH	3



21.- Número de objetos por cada de segmento en cada Tablespace.

Comentario: número de objetos por cada de segmento en cada Tablespace.

TABLESPACE_NAME||''||SEGMENT_TYPE	OBJETOS
AAS11_INDICES INDEX	3
AAS11_TABLAS INDEX	12
AAS11_TABLAS LOBINDEX	1
AAS11_TABLAS LOBSEGMENT	20
AAS11_TABLAS TABLE	15